WL-TR-95-3006

ASTROS ENHANCEMENTS

Volume III — ASTROS THEORETICAL MANUAL

D.J. NEILL
D.L. HERENDEEN

Universal Analytics, Inc.
3625 Del Amo Blvd. Suite 370
Torrance, CA 90503

V.B. VENKAYYA

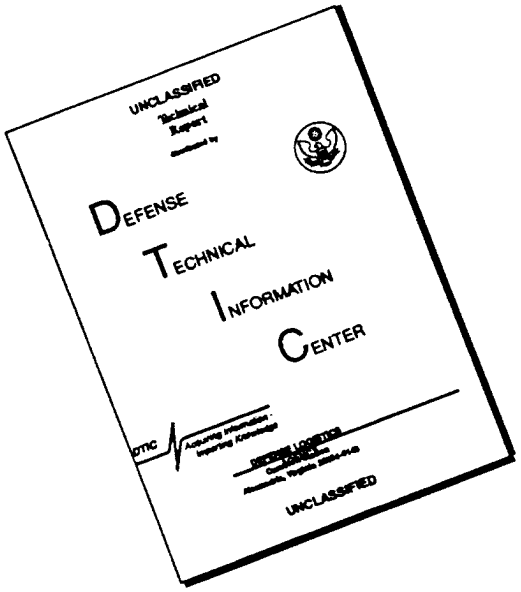
Air Force Wright Laboratory
Structures Division
Design Development Branch
Wright-Patterson Air Force Base, OH 15433-7542
May 1995

FINAL REPORT FOR THE PERIOD JANUARY 1987 - APRIL 1995

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED

FLIGHT DYNAMICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-7562

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

NOTICE

WHEN GOVERNMENT DRAWINGS, SPECIFICATIONS, OR OTHER DATA ARE USED FOR ANY PURPOSE OTHER THAN IN CONNECTION WITH A DEFINITE GOVERNMENT-RELATED PROCUREMENT, THE UNITED STATES GOVERNMENT INCURS NO RESPONSIBILITY OR ANY OBLIGATION WHATSOEVER. THE FACT THAT THE GOVERNMENT MAY HAVE FORMULATED OR IN ANY WAY SUPPLIED THE SAID DRAWINGS, SPECIFICATION, OR OTHER DATA. IS NOT TO BE REGARDED BY IMPLICATION, OR OTHERWISE IN ANY MANNER CONSTRUED, AS LICENSING THE HOLDER, OR ANY OTHER PERSON OR CORPORATION; OR AS CONVEYING ANY RIGHTS OR PERMISSION TO MANUFACTURE, USE, OR SELL ANY PATENTED INVENTION THAT MAY IN ANY WAY BE RELATED THERETO.

THIS REPORT IS RELEASABLE TO THE NATIONAL TECHNICAL INFORMATION SERVICE (NTIS). AT NTIS, IT WILL BE AVAILABLE TO TEH GENERAL PUBLIC, INCLUDING FOREIGN NATIONS.

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.

Raymond M. Kolonay
Project Engineer
Design & Methods Development Section

Nelson D. Wolf, Technical Manager
Design & Methods Development Section
Design Development Branch

George R. Holderby, Chief
Design Development Branch
Structures Division

IF YOUR ADDRESS HAS CHANGED, IF YOU WISH TO BE REMOVED FROM OUR MAILING LIST, OR IF THE ADDRESSEE IS NO LONGER EMPLOYED BY YOUR ORGANIZATION, PLEASE NOTIFY WL/FIBAD BLDG 45, 2130 EIGHTH STREET, SUITE 1, WRIGHT-PATTERSON AFB OH 45433-7552 TO HELP MAINTAIN A CURRENT MAILING LIST.

Copies of this report should not be returned unless return is required by security consideration, contractual obligations, or notice on a specified document.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	MAY 1995	FINAL 01/15/87--04/30/95

4. TITLE AND SUBTITLE
ASTROS ENHANCEMENTS
VOLUME III - THEORETICAL MANUAL

5. FUNDING NUMBERS
CONTR: F33615-87-C-3216
PE: 62201F
PROJ: 2401
WORK UNIT: 82

6. AUTHOR(S)
D. J. NEILL
D. L. HERENDEEN
V. B. VENKAYYA

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
UNIVERSAL ANALYTICS INC.
3625 DEL AMO BLVD. SUITE 370
TORRANCE CA 90503

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
FLIGHT DYNAMICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT PATTERSON AFB OH 45433-7562

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
WL-TR-96-3006

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE: DIRSTRIBUTION IS UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

ASTROS (Automated STRuctural Optimization System) is a computer program for the multidisciplinary design and analysis of aerospace structures. ASTROS combines mathematical optimization algorithms with traditional structural analysis disciplines such as static forces, normal modes, static aeroelasticity, and dynamic aeroelasticity (flutter), all in a finite element context, to perform automated preliminary design of an aircraft structure. This report is a complete user's manual that documents the many features of ASTROS through version 12 of the software package. It also provides information on system architecture and other topics of interest. This report is Volume 3 of a set; Volume 1 (WL-TR-93-3025) is the user's manual.

14. SUBJECT TERMS	15. NUMBER OF PAGES
ASTROS, OPTIMIZATION, AEROELASTICITY, FLUTTER, FINITE ELEMENT METHOD, STRUCTURAL DESIGN, AEROSPACE STRUCTURES	254
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

TABLE OF CONTENTS

1. INTRODUCTION 1

1.1. ASTROS CONCEPTS 1

1.2. ASTROS CAPABILITIES 2

1.3. DOCUMENTATION 3

4. MULTIDISCIPLINARY ANALYSIS AND DESIGN 23

4.1. INTRODUCTION 23

4.2. MULTIDISCIPLINARY OPTIMIZATION 24

4.3. Objective Function 26

4.4. Design Variables 26

4.4.1. Local Variables 26
4.4.2. Global Variables 28

4.5. Constraints 30

4.5.1. Strength Constraints 30
4.5.2. Stiffness Constraints 33
4.5.3. Buckling Optimization — The Unstiffened Plate 36
4.5.4. Buckling Optimization — The Column 41
4.5.5. Model Characteristic Constraints 42

4.6. SYNTHETIC FUNCTIONS IN OPTIMIZATION 45

4.7. SENSITIVITY ANALYSIS 46

4.7.1. Linear Design Variables 47
4.7.2. Nonlinear Design Variables 47

3. SYSTEM ARCHITECTURE 13

3.1. THE ASTROS EXECUTIVE SYSTEM 14

3.1.1. The Controller 14
3.1.2. High Level Memory 15
3.1.3. Execution Monitor 15
3.1.4. Input/Output Subsystem 15

3.2. THE DATABASE 15
3.2.1. Matrix Entities 16
3.2.2. Relational Entities 17
3.2.3. Unstructured Entities 18
3.3. THE DYNAMIC MEMORY MANAGER 18
3.4. THE USER INTERFACE 19
3.5. ENGINEERING MODULES 20
2. BASICS AND NOMENCLATURE 5
2.1. STRUCTURAL GEOMETRY MODELING 5
2.2. DISPLACEMENT SETS 6
2.2.1. Physical Sets 6
2.2.2. Dynamic Reduction Sets 6
2.2.3. Dynamic Analysis Sets 8
2.2.4. Unsteady Aerodynamic Sets 8
2.3. NOMENCLATURE 8
2.3.1. Matrices and Vectors 9
2.3.2. Subscripts 10
2.3.3. Superscripts 11
2.3.4. Optimization Nomenclature 11
5. FINITE ELEMENT MODELING 51
5.1. FINITE ELEMENTS 51
5.1.1. Concentrated Mass Elements 51
5.1.2. Scalar Elements 52
5.1.3. One-Dimensional Elements 53
5.1.4. Two-Dimensional Elements 57
5.1.5. Solid Elements 60
5.1.6. The General Element 61
5.1.7. Rigid Elements 62
5.2. APPLIED LOADS 64
5.2.1. Mechanical Loads 65
5.2.2. Gravity Loads 65
5.2.3. Thermal Loads 65
5.3. STRENGTH CONSTRAINTS 65
5.3.1. BAR Element 67
5.3.2. QDMEM1 Element 68

5.3.3. QUAD4 Element 69
5.3.4. ROD Element 70
5.3.5. SHEAR Panel 71
5.3.6. TRMEM Element 72

5.4. GLOBAL ASSEMBLY OF MATRICES 73

6. STATIC ANALYSIS 77

6.1. MATRIX EQUATIONS FOR STATIC ANALYSIS 77

6.2. CONSTRAINT EVALUATION 81

6.3. SENSITIVITY ANALYSIS 82

6.3.1. Displacement Constraints 82
6.3.2. Von Mises Stress Constraints 83
6.3.3. The Gradient Method 84
6.3.4. The Virtual Displacement Method 88

7. MODAL ANALYSIS 89

7.1. GENERALIZED DYNAMIC REDUCTION 89

7.1.1. Inertia Relief Shapes 90
7.1.2. Approximate Eigenvectors 92

7.2. THE EIGENANALYSIS METHODS 95

7.2.1. The Inverse Power Method 95
7.2.2. The Given's Method 96
7.2.3. The FEER Method 97

7.3. CONSTRAINT EVALUATION 100

7.4. FREQUENCY CONSTRAINT SENSITIVITIES 101

8. AERODYNAMIC ANALYSES 103

8.1. STEADY AERODYNAMICS 103

8.1.1. USSAERO Capabilities 103
8.1.2. USSAERO Methodology 105

8.2. UNSTEADY AERODYNAMICS 107

8.2.1. Unsteady Aerodynamics Capabilities 107
8.2.2. Unsteady Aerodynamics Methodology 108

8.3. CONNECTING AERODYNAMIC AND STRUCTURAL MODELS 111

8.3.1. Surface Spline 111
8.3.2. Linear Spline 113

8.3.3. Attachment of Splines with Elastic Springs . . . 116
8.3.4. Rigid Arms on Splines . . . 117
8.3.5. Equivalent Force Transfer . . . 118
8.3.6. Use of Splines . . . 119

9. STATIC AEROELASTIC ANALYSIS . . . 121

9.1. MATRIX EQUATIONS FOR STATIC AEROELASTIC ANALYSIS . . . 121

9.2. STABILITY DERIVATIVES . . . 127

9.2.1. Restrained and Unrestrained Stability Derivatives . . . 129

9.3. SYMMETRIC ANALYSES . . . 131

9.3.1. Trim Analysis . . . 131
9.3.2. Lift Effectiveness Constraint . . . 132
9.3.3. Flexible Stability Derivatives . . . 133
9.3.4. Trim Parameter Constraint . . . 134

9.4. ANTISYMMETRIC ANALYSES . . . 134

9.4.1. Trim Analysis . . . 134
9.4.2. Aileron Effectiveness Constraint . . . 134
9.4.3. Flexible Stability Derivative Constraint . . . 135
9.4.4. Trim Parameter Constraint . . . 135

9.5. SENSITIVITY ANALYSIS . . . 135

9.5.1. Trim Sensitivity Analysis . . . 138
9.5.2. Lift Effectiveness Sensitivity . . . 138
9.5.3. Aileron Effectiveness Sensitivity . . . 139
9.5.4. Flexible Stability Derivative Sensitivities . . . 140
9.5.5. Trim Parameter Sensitivity . . . 141

10. FLUTTER ANALYSIS . . . 143

10.1. THE P-K FLUTTER ANALYSIS . . . 143

10.2. FLUTTER CONSTRAINT EVALUATION . . . 147

10.3. SENSITIVITY OF FLUTTER CONSTRAINTS . . . 148

11. DYNAMIC ANALYSIS . . . 153

11.1. DYNAMIC MATRIX ASSEMBLY . . . 153

11.1.1. Direct Matrix Input . . . 155
11.1.2. Reduction of Direct Matrix Input . . . 155
11.1.3. Damping Options . . . 156

11.2. DYNAMIC LOADS GENERATION . . . 157

11.2.1. Transient Loads . . . 157
11.2.2. Frequency Dependent Loads . . . 158
11.2.3. Gust Loads . . . 159

11.3. TRANSIENT RESPONSE ANALYSIS . . . 160

11.3.1. Solution of Uncoupled Transient Response Equations . . . 160
11.3.2. Solution of Coupled Transient Response Coupled Equations . . . 162
11.3.3. Solution of Transient Equations Using Fast Fourier Transforms . . . 164

11.4. FREQUENCY RESPONSE ANALYSIS . . . 164

11.4.1. Solution of Uncoupled Frequency Response Equations . . . 165
11.4.2. Solution of the Coupled Frequency Response Equations . . . 165
11.4.3. Solution of Frequency Response Equation Including Gusts . . . 165

12. NUCLEAR BLAST RESPONSE . . . 167

12.1. INTRODUCTION . . . 167

12.2. THE AERODYNAMICS PREPROCESSOR . . . 167

12.3. TRIM FOR THE BLAST ANALYSIS . . . 170

12.4. BLAST RESPONSE . . . 171

13. AUTOMATED DESIGN . . . 175

13.1. MATHEMATICAL PROGRAMMING . . . 175

13.1.1. Reduction of the Number of Design Variables . . . 176
13.1.2. Reduction of the Number of Constraints . . . 176
13.1.3. The Approximate Design Problem . . . 177
13.1.4. Termination Criteria . . . 180

13.2. FULLY STRESSED DESIGN . . . 181

13.2.1. The FSD Algorithm for Local Design Variables . . . 182
13.2.2. Global Design Variable Determination . . . 183

14. REFERENCES . . . 185

APPENDIX A. THE QUAD4 ELEMENT . . . 187

A.1. DISPLACEMENT FUNCTIONS . . . 187

A.2. STRAIN-DISPLACEMENT RELATIONSHIP . . . 189

A.3. STRESS-STRAIN RELATIONSHIPS . . . 196

A.4. STIFFNESS MATRIX . . . 202

A.5. CONSISTENT AND LUMPED MASS MATRICES 202
A.6. STRESS RECOVERY 203
A.7. FORCE RESULTANTS 204
A.8. THERMAL LOAD VECTOR 205
A.9. LAMINATED COMPOSITE MATERIALS 206
A.9.1. Overview of Theory 206
A.9.2. Element Layer Stress Recovery 213
A.9.3. Interlaminar Shear Stresses 214
A.9.4. Force Resultants 214
A.9.5. Failure Indices 215
A.10. CORRECTION OF OUT-OF-PLANE SHEAR STRAIN 217
A.10.1. Geometric Variables 218
A.10.2. Edge Shears and Shear Vectors 219
A.10.3. Nodal Contributions of Shear Strain 220
A.10.4. Transformations 221
APPENDIX B. NUCLEAR BLAST ANALYSIS 223
APPENDIX C. THE FAST FOURIER TRANSFORM 229
C.1. DISCRETE FOURIER TRANSFORM 230
C.2. FAST FOURIER TRANSFORMS 231
C.3. IMPLEMENTATION CONSIDERATIONS 233

LIST OF FIGURES

Figure 1. Hierarchy of Displacement Sets 7
Figure 2. Relation of Dynamic Analysis Sets 8
Figure 3. The ASTROS System Architecture 13
Figure 4. The Packed Matrix Format 16
Figure 5. Example of a Relational Entity 17
Figure 6. An Unstructured Data Entity 18
Figure 7. Dynamic Memory 19
Figure 8. Multidisciplinary Optimization 25
Figure 9. Intermediate Complexity Wing Structure 29
Figure 10. Buckling Panel 37
Figure 11. The ROD Element 53
Figure 12. The BAR Element 55
Figure 13. The PBAR1 Standard Sections 56
Figure 14. The Quadrilateral Shear Element 57
Figure 15. The Triangular Membrane Element 58
Figure 16. The Quadrilateral Membrane Element 59
Figure 17. The TRIA3 Element Geometry 59
Figure 18. The QUAD4 Element Geometry 59
Figure 19. Aerodynamic Paneling in USSAERO 104
Figure 20. Application of ATTACH Option 118
Figure 21. Flutter Analysis Algorithms Within ASTROS 146
Figure 22. Block Diagram for the Nuclear Blast Calculation 168
Figure A–1. Internal Element Coordinate System 188
Figure A–2 Isoparametric Quadrilateral 4-Node 188

LIST OF FIGURES (Continued)

Figure A-3. Deformations at Grid Point i 190
Figure A-4. Deformations in the Global Direction 192
Figure A-5. Material and User Defined Element Axes 197
Figure A-6. Geometry of an N-Layered Element 208
Figure C-1. CPU Time Comparison 231

LIST OF TABLES

Table 1. Matrix Nomenclature 9
Table 2. Subscript Nomenclature 10
Table 3. Superscript Nomenclature 11
Table 4. Optimization Nomenclature 11
Table 5. Large Matrix Utilities 21
Table 6. Physical Design Variables 27
Table 7. Summary of ASTROS Rigid Elements 62
Table 8. Structural Element Design Constraints 66
Table 9. Aerodynamic Panel Points 109

THIS PAGE INTENTIONALLY LEFT BLANK

FOREWORD

The Analysis and Optimization Branch of the Air Force Wright Laboratory initiated the development of the Automated Structural Optimization System (ASTROS) in 1983.This final report is submitted in fulfillment of CDRL CLIN 0003, Sequence Number 13, of the ASTROS Enhancement Contract, F33615-87-C-3216, dated 29 January 1987. This report, which is one of a three volume final report is the Theoretical Manual that describes the analytical foundations for ASTROS.

This work was performed by Universal Analytics, Inc. and their subcontractor, Northrop Corp. This manual supercedes the original ASTROS Theoretical Manual documented in AFWAL-TR-88-3028, Volume I by E.H. Johnson and V.B. Venkayya. The major contributors to this report were D.J. Neill, the Associate Program Manager, D.L. Herendeen, the Program Manager, and Dr. V.B. Venkayya of the Flight Dynamics Directorate of the Wright Laboratory. E.H. Johnson, previously of Northrop, and Dr. J. R. Ruetenik, of Kaman AviDyne, were major contributors to the original report.

Capt. R. Canfield, Capt. S. Pitrof and R. Kolonay have been the Air Force Program Engineers, and Dr. V. B. Venkayya initiated the program and has provided much of the overall program direction.

Additional contributors to the original report included Dr. H. Chang, R.L. Hoesly, and J.Jalil of Universal Analytics Inc., R.E. Blauvelt of Northrop, and R.F. Smiley and G. Tremainis of Kaman AviDyne. Additional technical guidance was provided by Dr. W. P. Rodden and Dr. K. Appa.

1. INTRODUCTION

The design of aircraft and space structures requires the marshaling of large teams of engineers to select a design which satisfies all requirements. Typically this design goes through further refinement or modification as more knowledge is gained about requirements or as new conditions are imposed. Much of this effort presently consists of applying laborious "cut and try" procedures wherein the design is perturbed and reanalyzed many times. This redesign frequently is required because two or more disciplines have conflicting demands that require compromise.

The goal of Air Force Contract F33615-87-C-3216, and its precursor, Contract F33615-C-83-3232, has been to provide an automated design and analysis tool that performs the tradeoff and synthesis tasks in a systematic way. The ASTROS (Automated STRuctural Optimization System) software system has resulted from this effort and this Theoretical Manual, plus the companion User's Manual, and Applications Manual, provide the information required to understand, apply and modify the software. This introduction provides a broad overview of ASTROS concepts and capabilities, discusses the contents of ASTROS documentation and provides information on supplementary references.

1.1. ASTROS CONCEPTS

ASTROS is a finite element-based software system that has been designed to assist, to the maximum practical extent, in the preliminary design of aerospace structures. A concerted effort has been made to provide the user with a tool that has general capabilities with flexibility in their application.

A vital consideration in software of this type is that the key disciplines that impact the design must be included in the automated design task. This multidisciplinary aspect of the program has been implemented in an integrated way so that all the critical design conditions are considered simultaneously.

In addition to the interaction of several disciplines, ASTROS can treat multiple boundary conditions and, within each boundary condition, multiple subcases. The system is not arbitrarily restricted by problem size, and it conforms to the current environment for performing structural analysis in the aerospace industry. The practical limitations on problem size are available disk space and data processing time.

ASTROS supports the multidisciplinary nature of design by implementing the disciplines in separate modules and by the use of ***MAPOL*** (Matrix Analysis Problem Oriented Language), a high level language, to direct the interactions among the modules. Data transfer is accomplished using the ***CADDB*** (Computer Automated Design Database) that has also been developed for this project.

The requirement for large problem size is addressed by the presence of a ***Dynamic Memory Manager*** that allocates memory in a way that eliminates the need for fixed length arrays. Allocations are made and destroyed dynamically so that free memory can be shared by the engineering modules.

Finally, compatibility with the current aerospace environment is addressed because the ASTROS procedures resemble those of NASTRAN in terms of user input and pre- and post-processor interfaces. While the ASTROS program does not contain many of the specialized capabilities available in NASTRAN, the basic structural analysis features have been included. Most importantly, from a user point-of-view, the Bulk Data formats have been taken directly from NASTRAN and modified only if the design considerations required such a modification in the data or, in a few cases, if minor changes result in superior capability. New Bulk Data entries have been created to input design information and data needed to run the steady aerodynamics and other analyses specific to ASTROS.

1.2. ASTROS CAPABILITIES

The balance of this manual documents the engineering analyses within ASTROS. This section gives a brief overview of the capabilities that are included in the code. The basic disciplines that are implemented within this code are as follows:

1. Static analysis
2. Modal analysis
3. Aerodynamic Analysis
4. Dynamic Analysis
5. Optimization

The statics analysis methodology is based on a finite element representation of the structure, as are all the structural analysis disciplines in ASTROS. The static analyses compute responses to statically applied mechanical (e.g., discrete forces and moments), thermal and gravity loadings. Static deformations and their resultant stresses are among the computed responses. An extensive design capability is provided for the static analysis discipline. Details of this discipline are provided in Section 6 of this report.

The modal analysis capability in ASTROS permits the determination of a structure's eigenfrequencies and normal modes. As outlined in Section 7 of this report, the reduction of the finite element model to a size tractable for performing an eigenanalysis is performed by one of two techniques. In the first, the degrees of freedom are reduced to a user specified analysis set through the use of Guyan reduction. The second technique employs Dynamic Reduction concepts to produce basis vectors that are "rich" in the eigenvectors of the structure. The design capability for modal analysis is limited to the ability to impose limits on the natural frequencies of the structure. Apart from its inherent usefulness, the modal analysis capability also serves as the basis for further analyses, such as flutter, transient response and frequency response, that can be performed using modal coordinates.

The aerodynamic analyses in ASTROS include both steady and unsteady formulations. These could be considered as separate disciplines, but they are linked in this report because of the fact that they share the method for linking quantities computed in the aerodynamic models to the structural model. Section 8 first discusses these spline techniques and then separately discusses the steady and unsteady

aerodynamic analyses. Section 9 discusses the use of the steady aerodynamics to provide loads on a free flying aircraft for specified longitudinal flight conditions and to provide estimates of the rolling effectiveness of control surfaces in antisymmetric maneuvers. All the design conditions that can be applied to a static analysis can also be imposed on the symmetric flight condition. In addition, limits on the aircraft's lift effectiveness and rolling effectiveness can be imposed.

The unsteady aerodynamics are used for flutter, gust and nuclear blast analyses. Section 10 provides a description of the algorithms used to perform flutter analysis and design. Flutter design requirements are specified in terms of the required damping levels at user specified velocities.

The dynamic analysis disciplines listed above represent a breadbasket of methods that are detailed in Sections 11 and 12. These methods share the characteristic that they include time or frequency varying loads as well as inertial terms (i.e., those proportional to the structure's acceleration) and optional damping terms (i.e., those proportional to the structure's velocity). Section 11 discusses transient and frequency analyses that utilize either a direct or a modal representation of the structure while Section 12 discusses the specialized dynamic response of an aircraft to a nuclear blast. All the dynamic analyses in ASTROS share the property that only an analysis capability, with no design conditions, is provided. The rationale for including these further analyses, in what is basically a structural design procedure, is that it allows the user to check the final achieved design for a variety of other conditions within the context of ASTROS. This is in contrast to requiring the user to understand and develop models for a series of more specialized procedures.

The final discipline listed above is that of optimization. If only stress, or strain, constraints are included in the design task, the fully stressed design option may be used. For more general design tasks, a mathematical programming approach has been implemented. Section 13 discusses both of these methods and provides details on the extensive use of approximation concepts to make the design task tractable when many design variables and design conditions are used.

1.3. DOCUMENTATION

This subsection provides a brief description of each of the ASTROS documentation manuals, as well as other references that are central to the ASTROS procedure. The ASTROS documentation is divided into the following four manuals:

1. VOLUME I Theoretical Manual
2. VOLUME II User's Manual
3. VOLUME III Applications Manual
4. VOLUME IV Programmer's Manual

This Theoretical Manual contains theoretical background on both the computer science and engineering analyses of the ASTROS system. Emphasis is given to the more innovative aspects of the ASTROS system while relying on other sources to detail those features that are common to other procedures.

The User's Manual contains the information needed to run the ASTROS system. The user input is documented, as is information on the output quantities that can be computed. The user is also provided

with information on how to modify the standard MAPOL sequence or to write a specialized MAPOL program to tailor ASTROS to a particular application.

The Applications Manual serves a number of functions. The first is to describe, in some detail, alternate sources of information. Secondly, it provides guidelines and modeling information on the use of more unique features of the procedure. For example, the steady aerodynamic and design capabilities are discussed in some detail since these are unique to ASTROS. Finally, the Applications Manual contains a number of sample runs that can be used to check out the initial installation of the procedure and further guide ASTROS usage. Note that the Applications Manual was created under the original contract effort, and it has not been updated under the ASTROS Enhancement contract.

The Programmer's Manual is reserved for researchers who wish to make modifications to the ASTROS code, either to insert a new module or to modify an existing capability. A large percentage of this manual is the documentation of the database entities. Other useful sections of the report are the definitions of the calls to utility routines. Also, the installation of the procedure on different machines is presented for the "system administrator."

In terms of subsidiary documentation, ASTROS relies heavily on NASTRAN in terms of methodology and as a starting point for code development. NASTRAN documentation, therefore, is useful in understanding ASTROS. As mentioned, the ASTROS documentation, and particularly the Theoretical Manual, emphasizes the more novel aspects of the ASTROS code while relying on this other documentation for the more standard features. For example, this theoretical manual contains no description of the large matrix utilities while the NASTRAN Manual of Reference 1 devotes 21 pages to these utilities. This reliance is less evident in the other manuals. The ASTROS Programmer's Manual is considerably more succinct than the corresponding NASTRAN manual of Reference 2 in terms of module definition, but does provide some documentation for each module. The ASTROS User's Manual is intended to be standalone and is sufficiently different from the corresponding manual of Reference 3 that one is advised not to rely too heavily on preconceptions based on using NASTRAN. On the other hand, the similarities between ASTROS and NASTRAN inputs are so marked that it should be extremely easy for a user to go from one system to another. The Applications Manual discusses in greater detail how other reference sources can be used to supplement the ASTROS documentation.

2. BASICS AND NOMENCLATURE

This section provides definitions of basic structural analysis terms as they are used in ASTROS. Recall that ASTROS concepts and notation follow those used in NASTRAN to the maximum extent possible. With the exception of subsection 2.5, which describes the notation used for sensitivity and optimization equations, the contents of this section should be familiar to a typical NASTRAN user and are redundant with existing NASTRAN documentation, such as that found in References 1 and 14. It is provided here because the use of coordinate systems and displacement sets are pervasive in the remainder of this manual and it is therefore necessary to have clear definitions of them.

2.1. STRUCTURAL GEOMETRY MODELING

The geometry of the structural model is defined by the user in terms of ***grid points*** and ***scalar points***. Grid points are located in space by user defined coordinates and each point has six ***degrees of freedom***. Scalar points have a single degree of freedom that has no geometric definition but is included in the solution set. Scalar points are used to conveniently include scalar elements, such as springs and mass elements, in the structural representation.

The geometry definitions are made in terms of ***coordinate systems***. To simplify input, the user is permitted to define any number of coordinate systems in the bulk data packet. ASTROS then rationalizes these systems into a single system for performing the analyses. The input (and output) coordinates can be specified in terms of ***rectangular***, ***cylindrical*** or ***spherical*** systems. The concepts of Local, Global and Basic coordinate systems also need to be understood in order to prepare ASTROS input and interpret the results.

A ***Local coordinate system*** is one that is chosen for convenience in specifying element geometries. A given structure is typically divisible into components and surfaces that naturally present themselves. Each of these is modeled most efficiently through the use of a local coordinate system.

The ***Global coordinate system*** is the single system in which the structural analysis is performed and the results are presented. It should be emphasized that this coordinate system is not necessarily defined by a single axis system. Instead, it is the collection of all the user specified output coordinate systems.

The ***Basic coordinate system*** is the single system relative to which all other systems are defined. In this case, it can be depicted by a single axis system and it is necessary that all geometric points be able to be defined in this coordinate system before ASTROS can proceed. This definition is done internally and the user has no need to be aware of the computations required to get the coordinates into this system.

2.2. DISPLACEMENT SETS

ASTROS has maintained the NASTRAN terminology in defining displacement sets in structural analysis. This discussion introduces these sets since their definition is required in all the disciplines described in the remainder of this report. The following sections discuss the four classes of sets: physical, dynamic reduction, dynamic analysis and unsteady aerodynamic .

2.2.1. Physical Sets

The term ***physical*** refers to those sets whose members have a specific physical meaning and are related directly to the degrees of freedom in the analysis. Figure 1 depicts the hierarchy of sets that are used in the standard static and modal analysis disciplines described in Sections 6 and 7.

Starting at the top of the figure, the *g-set* contains all the degrees of freedom in the structural model. The size of this set is equal the number of scalar points plus six times the number of grid points. This set can be divided into one set (the *m-set*) whose members are defined to be explicitly dependent on the second, independent set (the *n-set*). These dependencies are designated multipoint constraints.

At the next level of division, the *n-set* degrees of freedom are divided into those whose displacements are user specified (the *s-set*) and those that are left free for solution (the *f-set*). The specified displacements are most typically used to constrain rigid body motions, either by setting degrees of freedom with no associated stiffness to zero or by applying fixity conditions at the structure's boundary. They can also be used to force the structure to deform to certain user specified values. It is useful to make a distinction between those degrees of freedom that are constrained for all boundary conditions (permanent single point constraints) and those that may be boundary condition dependent.

The next reduction divides the *f-set* into the omitted (the *o-set*) and the analysis (the *a-set*) degrees of freedom. This reduction is done primarily to make a modal analysis task tractable and has less utility for a static analysis. The multidisciplinary nature of ASTROS, however, makes it desirable to use an *a-set* in a static analysis if the same boundary condition also requires dynamic analyses. The selection of degrees of freedom for the two sets is somewhat arbitrary and therefore puts a burden on the user. Dynamic Reduction, discussed in Subsection 7.1, is an attractive alternative to this selection process.

If the structure has ***rigid body degrees of freedom***, such as a complete aircraft or spacecraft, a further reduction is required before the static response can be obtained. In this reduction, the *a-set* is divided into a set that is just sufficient to remove the rigid body motions (the reference or *r-set*) and a set of remaining (the left over or *l-set*) degrees of freedom.

2.2.2. Dynamic Reduction Sets

The ***Dynamic Reduction technique*** of Subsection 7.1 defines two further sets. The first is a set of generalized degrees of freedom for the approximate eigenvectors of the reduction process and is designated the *j-set*. The second is a set of generalized degrees of freedom for the inertia relief shapes and is designated the *k-set*.

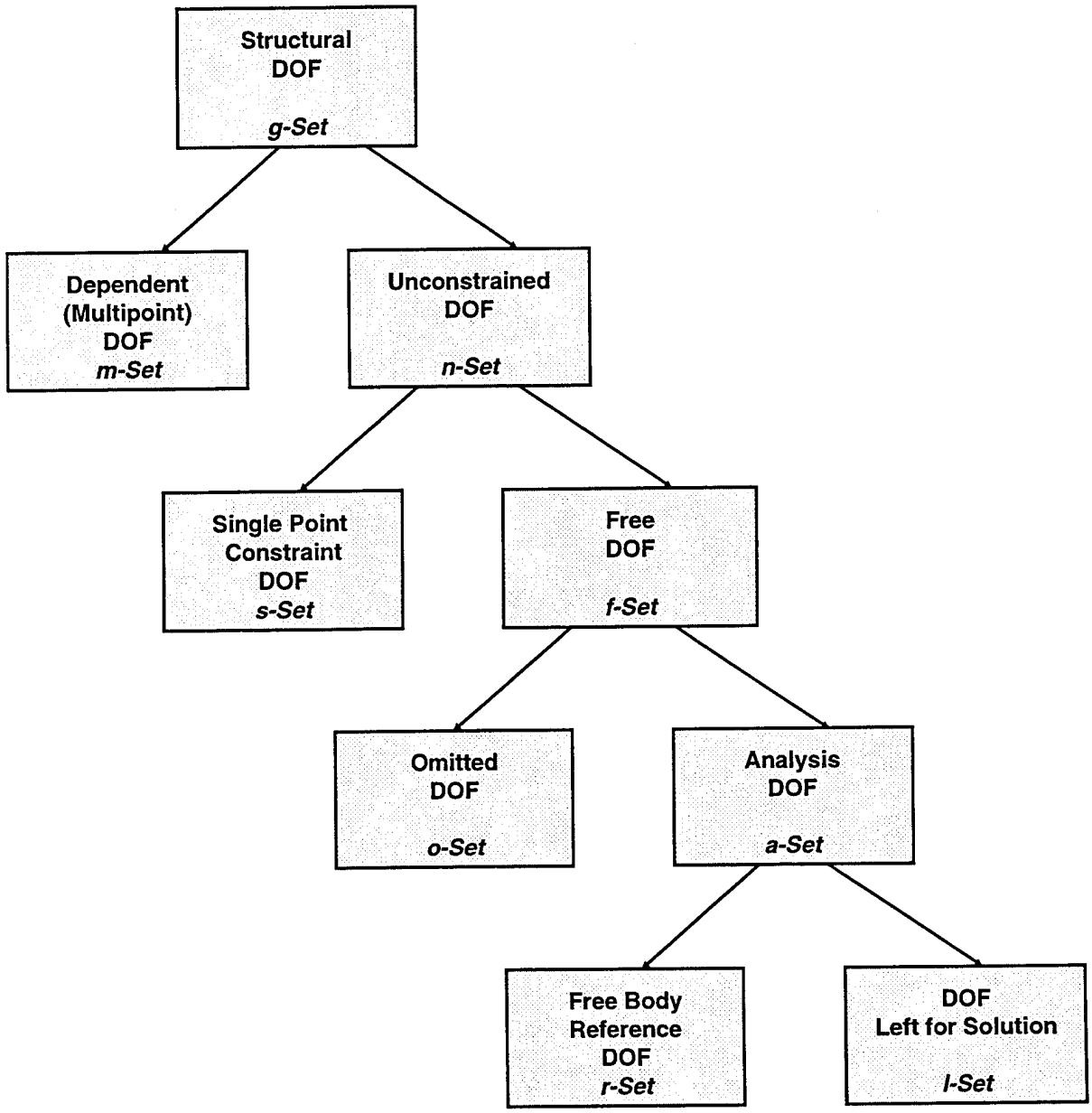


Figure 1. Hierarchy of Displacement Sets

2.2.3. Dynamic Analysis Sets

Modal analyses produce ***generalized coordinates*** that represent further sets that are used in subsequent dynamic analyses, such as flutter and frequency response. In addition, the representation of control systems is effected through the definition of ***extra points*** that make up a further set. The sets involved in dynamic analysis are shown in Figure 2. The set of generalized coordinates associated with the eigenvectors determined in a modal analysis is designated the *i-set*. The extra point degrees of freedom are contained in the *e-set* and the union of these two sets is the *h-set*. Dynamic analyses performed directly in the physical degrees of freedom utilize the *d-set*, which is the union of the *e-set* and the *a-set*. A final set, which is in addition to those shown in Figures 1 and 2, is the union of the *g-set* and the *e-set* and it constitutes the complete physical degrees of freedom (the *p-set*).

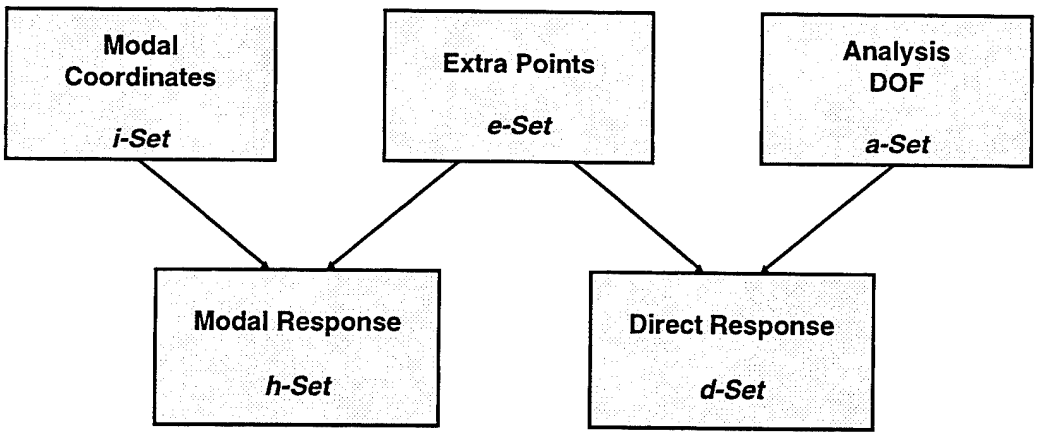


Figure 2. Relation of Dynamic Analysis Sets

2.2.4. Unsteady Aerodynamic Sets

As discussed in Section 7, ASTROS aerodynamic models are independent of the structural model and therefore have their own degrees of freedom. For the unsteady aerodynamics model, ASTROS follows the NASTRAN convention and refers to these degrees of freedom as the *k-set* and uses the *j-set* to refer to the aerodynamic boxes. Note that the *j-set* and *k-set* have been defined in a different manner in Subsection 2.2.2 and the appropriate definition must be determined from context.

2.3. NOMENCLATURE

Standard nomenclature, as used in structural analysis literature in general and NASTRAN in particular, has been adopted to the maximum extent possible. This section defines matrix and subscript notation as it is used throughout the balance of this report. This is not a comprehensive list, with additional definitions for specialized notation provided where the term is first used. The standard

MAPOL sequence also conforms to this notation, with the limitation that subscripting is not available in the MAPOL language so that matrix names and their subscripts make up the MAPOL name (e.g., $\boldsymbol{M}_{aa}$ in this document becomes **MAA** in the MAPOL sequence).

2.3.1. Matrices and Vectors

Matrices and vectors in the report are denoted by bold italicized type. The matrices defined in Table 1 are typically subscripted to indicate the set to which the matrix is referred (e.g., the $\boldsymbol{M}_{aa}$ matrix just discussed is in the *a-set* while the $\boldsymbol{M}_{oa}$ matrix has rows associated with the *o-set* and columns associated with the *a-set*).

Table 1. Matrix Nomenclature

TERM	(M)ATRIX or (V)ECTOR	DESIGNATION
$\boldsymbol{B}$	M	Damping
$\boldsymbol{D}$	M	Rigid body transformation
$\boldsymbol{G}$	M	Transformation matrix, including spline matrices for steady aerodynamics
$\boldsymbol{K}$	M	Structural stiffness
$\boldsymbol{M}$	M	Mass
$\boldsymbol{m}$	M	Rigid body mass
$\boldsymbol{P}$	V/M	Applied load
$\boldsymbol{u}$	V/M	Displacement
$\boldsymbol{UG}$	M	Unsteady aerodynamic spline
$\boldsymbol{YS}$	V	Enforced displacements

2.3.2. Subscripts

The subscripts listed in Table 2 correspond, in most cases, to the displacement sets discussed in Subsection 2.2. Certain of the subscripts are seen to have multiple definitions and the appropriate definition will either be clear from context or defined explicitly in the text.

Table 2. Subscript Nomenclature

SUBSCRIPT	DEFINITION
a	Analysis set
d	Dynamic set
e	Extra point set
f	Free set
g	Global set
h	Modal analysis set
i	Modal coordinates set or Design variable identification
j	Inertia relief shape coordinates set or Constraint identification or Aerodynamic box set
k	Approximate eigenvector coordinates set or Aerodynamic set
l	Left over set
m	Multipoint constraint set
n	Independent set
o	Omitted set
p	Physical set
r	Rigid or support set
s	Single point constraint set

2.3.3. Superscripts

Table 3 presents a small set of superscripts that conform to those used in general structural analysis.

Table 3. Superscript Nomenclature

SUPERSCRIPT	DEFINITION
I	Imaginary part
R	Real part
T	Matrix transpose
-1	Matrix inverse
$\dot{x}$	(Single dot) Time first derivative or velocity
$\ddot{x}$	(Double dot) Time second derivative or acceleration
a	Aerodynamic terms are included

2.3.4. Optimization Nomenclature

Table 4 presents the most commonly found nomenclature used for sensitivity and optimization equations.

Table 4. Optimization Nomenclature

SYMBOL	DEFINITION
F	Objective function
$\boldsymbol{R}$	Right-hand-side of sensitivity equations
$\boldsymbol{t}$	Vector of local design variables
$\boldsymbol{v}$	Vector of global design variables
$\frac{\partial}{\partial t}$ or $\frac{\partial}{\partial v}$	Sensitivity coefficient
Δ	Finite difference step size
$\boldsymbol{A}$	Matrix of constraint sensitivities
$\boldsymbol{g}$	Vector of inequality constraint values

THIS PAGE INTENTIONALLY LEFT BLANK

3. SYSTEM ARCHITECTURE

A large, multidisciplinary software system such as ASTROS necessarily requires a flexible system architecture that serves as the basis for construction and integration of the software developed. The detailed specification of this architecture contains a significant computer science content that is not only outside the scope of this report, but also not of general interest. Nonetheless, the ASTROS user should have a basic familiarity with this architecture, since it permeates the implementation and application of the program. These basics are provided in this section, while details into particular aspects of the system design can be found in References 10, 11, and 12.

Figure 3 illustrates the components of the ASTROS architecture, emphasizing its modular form. An additional component that does not fit neatly on the figure is the Dynamic Memory Manager. Each of these components is discussed in the following subsections.

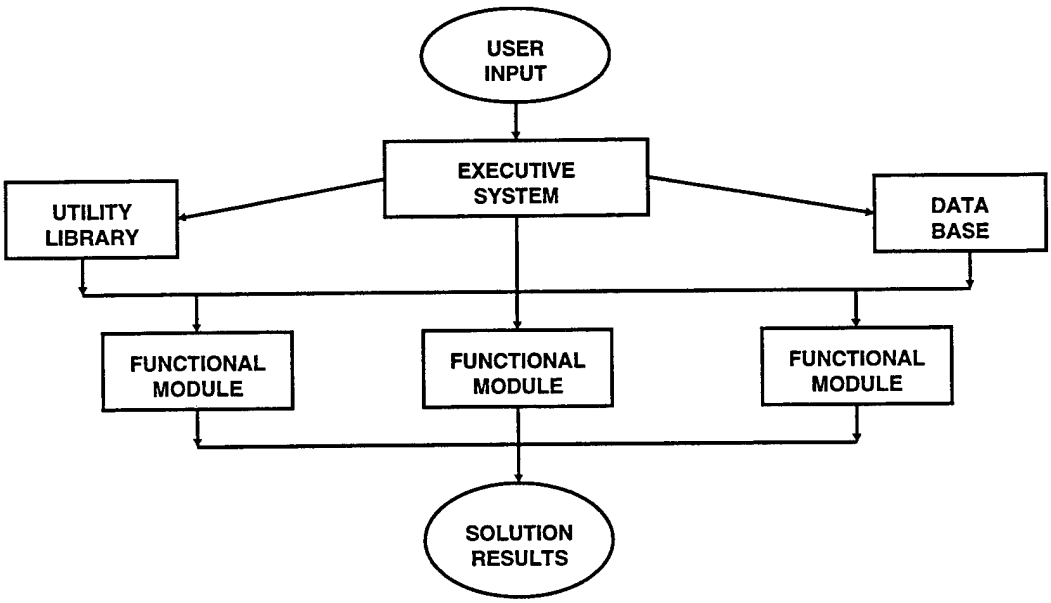


Figure 3. The ASTROS System Architecture

3.1. THE ASTROS EXECUTIVE SYSTEM

The ***Executive System*** is the heart of the software. It initiates the procedure, controls program flow and terminates execution. It is convenient to think of the Executive System as a stylized computer with four components found in an actual computer:

1. Control unit
2. High level memory
3. Execution monitor
4. Input/output subsystem

3.1.1. The Controller

The control unit, or controller, begins the execution. This is the routine that first performs standard initiation tasks, such as accommodating machine-dependent idiosyncrasies and initiating elapsed and CPU timers. Subroutines are also called which initialize the system and engineering databases and the dynamic memory manager. An initial pass is made through the input data stream, breaking it into five packets: Debug, MAPOL, Solution, Function, and Bulk Data. Information on the function and input requirements for each of these packets is given in the User's Manual, but the processing of the MAPOL packet needs to be further explained here in terms of how it effects the initiation and execution of the ASTROS system.

From the point of view of the user, ASTROS is driven by MAPOL (Matrix Analysis Problem Oriented Language). Such a control language, similar to the DMAP of NASTRAN or the typical query language of a database management system, has proven to provide maximum flexibility for the user. In particular, MAPOL provides features that include:

1. Structured, algorithmic language syntax
2. Special data types for matrices and relations
3. User-written procedures and an extendible procedure library
4. Complete run-time utility library
5. Embedded database operations.

Just as for any high level language, the translation of a MAPOL sequence from the actual input to the form used in controlling the execution is performed by a compiler. The MAPOL compiler creates two relations. The first, called MEMORY, is a map of the memory defined by the MAPOL program and discussed in the next subsection. The second relation, called MCODE, represents the executable code that performs operations directly and calls the functional modules within the ASTROS system.

Depending on user input, the controller operates on these two relations in one of three ways. If the user has selected the standard MAPOL sequence, there is no MAPOL packet and the MCODE and MEMORY relations contained in the system database are retrieved directly by the execution monitor. If the user has modified the standard sequence, an editing process takes place on the stored standard sequence. The edited sequence is then recompiled, replacing the data in the MCODE and MEMORY

relations. Finally, if the user has supplied a complete customized MAPOL sequence, the data in the two relations are replaced with new entries created by the MAPOL compiler.

3.1.2. High Level Memory

The MAPOL compiler reserves a space in memory for the ***ASTROS Run-time Memory***. (Note that this is separate from the MEMORY relation just discussed.) ASTROS run-time memory is of a *high-order*. This means that, unlike a normal computer memory, more than one word is used to store a data item. The ASTROS memory contains entries that are five single-precision computer words in length. The first word contains the data type and the next four words the actual memory contents. These contents may be integers, real values, in single or double precision, complex values, in single or double precision, or character data defining the names of database entities. Then, in a manner analogous to most machines, memory addresses are referenced by the executable code and modified during execution.

3.1.3. Execution Monitor

Following the initiation tasks discussed in above, the controller invokes the ***Execution Monitor*** to drive the ASTROS system. This monitor, using the instructions contained in the MCODE relation, directs the tasks specified in the MAPOL sequence. The monitor contains a processor which performs basic arithmetic and logical operations and also interfaces directly with a run-time library that performs simple mathematical and database operations. For more complex tasks, control is passed to the functional and utility modules discussed in Subsections 3.5 and 3.6.

3.1.4. Input/Output Subsystem

The executive system controls the files that are to be used for input and output. The principal I/O is performed by the Computer Automated Design Data Base (CADDB) as discussed in Subsection 3.2. The definition of FORTRAN logical units used for the user interface is also performed by the executive. Finally, a limited capability for sending data directly to the user output file is available from the MAPOL packet.

3.2. THE DATABASE

In a large scale engineering analysis system such as ASTROS, the efficient handling of the voluminous data required is a key element in the viability of the system. A specially designed database, called ***CADDB*** (***Computer Automated Design Data Base***), has been developed for the ASTROS system. The design of this database recognized the need for handling three distinct types of data. First, the structural analysis aspects of ASTROS impose a requirement for the storage and retrieval of very large, often sparse, matrices. A storage method is needed that minimizes disk storage requirements while allowing algorithms to be developed that can perform matrix operations of virtually unlimited size. The second requirement is the need to access individual data items directly and rapidly with minimum physical I/O. Such data items include the thickness of a single finite element or the data defining the properties of a particular material. Finally, there is a need to access heterogeneous collections of unstructured data very efficiently. This type typically represents working data which is generally used on an all-or-nothing basis within an individual module.

Existing available databases provide some, but not all, of these capabilities. In addition, many of these are commercial products with proprietary restrictions that are inconsistent with the basic ground rules for developing the ASTROS system. Therefore, a unique database was constructed which supports these three different representations. A significant benefit that accrued from this customized design was that a common structure was formulated for accessing the three types of data, i.e., a uniform, common applications interface has been provided to support each of the database entity classes. For example, a module may position to a specific matrix column, relational row, or unstructured record. This can then be followed by fetching all, or part, of the data stored at the current position.

Each of the three data types is now briefly described. Appendix 3 of Reference 12 contains more detailed information on CADDB and the Programmer's Manual contains applications interface information.

3.2.1. Matrix Entities

ASTROS is based upon the finite element method of structural analysis extended to include optimization. This method requires that all governing equations of motion be written in matrix form. This allows complex solutions to be performed using straightforward matrix algebra. Since the order of these matrices may be very large, it is essential that they be stored in a compressed, or ***packed***, format. This format exploits the strongly banded nature of most structural matrices — the low density of nonzero terms in these matrices allows enormous saving of storage space.

The packed format of matrices is shown in Figure 4. There are actually two levels of data compression. Firstly, any null column in a matrix is completely omitted. This extension to previous

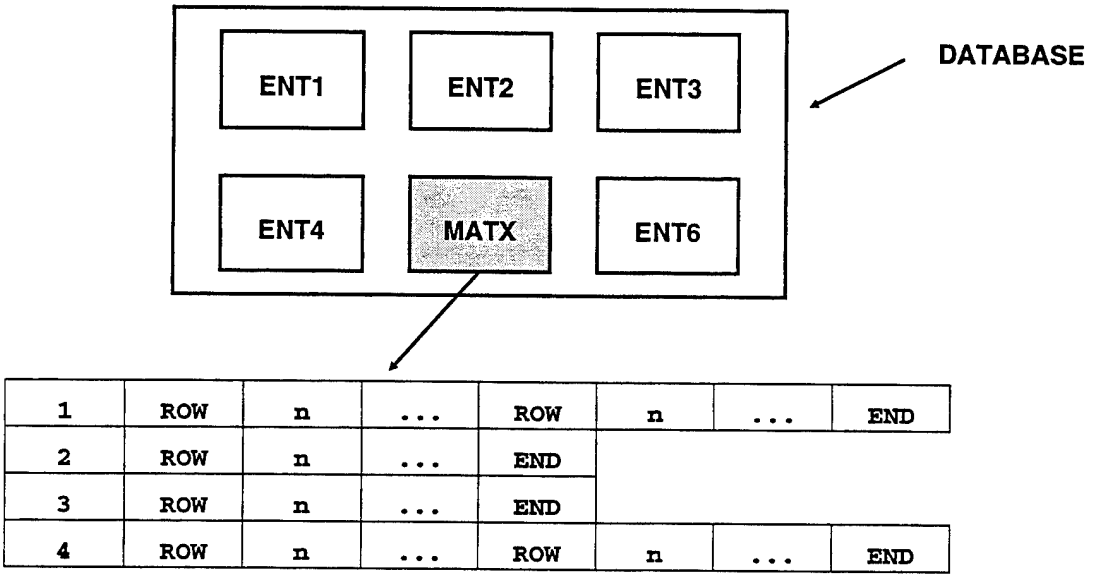
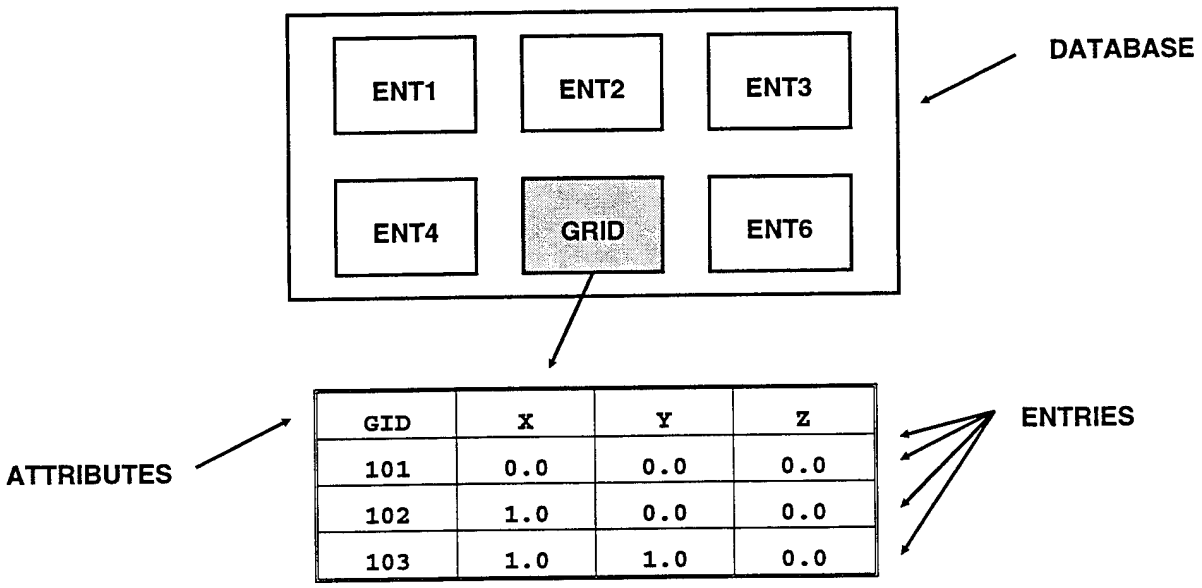


Figure 4. The Packed Matrix Format

methods of packing is well suited to the extremely sparse matrices arising from sensitivity calculations. Secondly, only strings of nonzero terms in a non-null column are actually stored. Each string contains a two word "header" which specifies the row position of the first nonzero term followed by the number of terms appearing. The header is then followed by the actual numeric values. This method of storage, pioneered by NASTRAN, has proven to be very effective.

3.2.2. Relational Entities

Relational Entities are essentially tables. The formalization of this type of data in recent years has found relevance across a wide variety of data processing applications (Reference 13). Each relation has rows, called ***entries*** and columns, called ***attributes***. Each attribute is given a descriptive name, a data type, and constraints on the values that the attribute may assume. These definitions are referred to as the ***schema*** of the relation. An example of a relation defining grid point data is shown in Figure 5. The importance of relational data to design optimization is that a single entry may be directly accessed based on qualified values of one or more of its attributes. This minimizes the actual I/O transfer required when modifying small amounts of data. CADDB further extends this capability by allowing a mechanism for rapidly accessing all of the data in a relation, if such access would be more efficient.



GID	X	Y	Z
101	0.0	0.0	0.0
102	1.0	0.0	0.0
103	1.0	1.0	0.0

Figure 5. Example of a Relational Entity

3.2.3. Unstructured Entities

There are many times that a software module requires temporary, or scratch, disk space while performing its task. These data are generally highly local and will not be passed to other modules within the system. To accommodate this requirement effectively, CADDB supports an unstructured entity type composed of ***records*** containing any arbitrary collection of data as shown in Figure 6. Once again, CADDB has the capability to directly access each of the records within the entity.

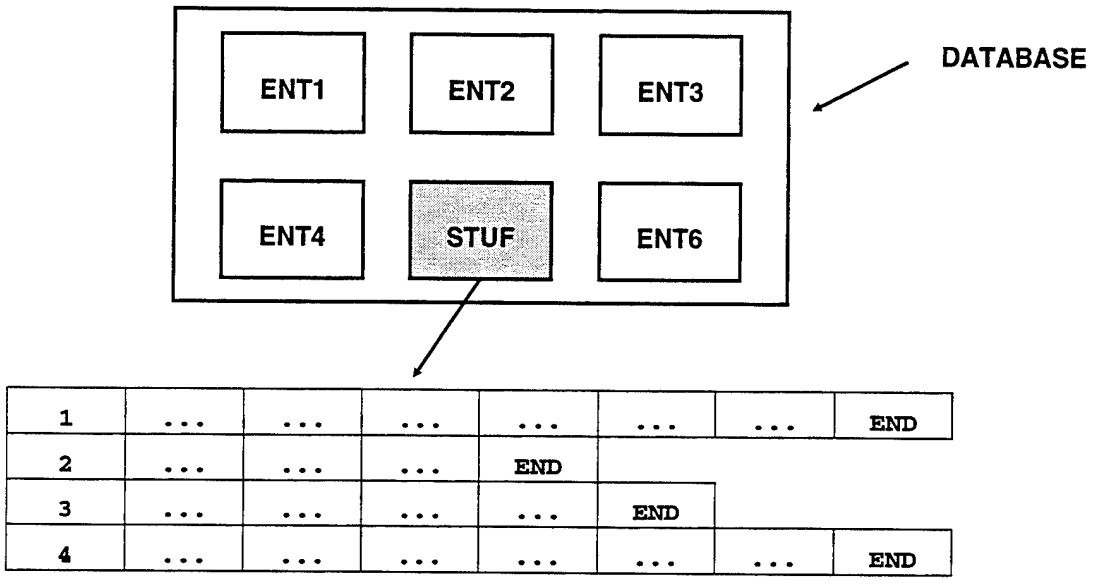


Figure 6. An Unstructured Data Entity

3.3. THE DYNAMIC MEMORY MANAGER

A key feature of the ASTROS system, that is not shown in Figure 3, is the ***Dynamic Memory Manager***. This feature allows modules to be written without resorting to fixed size arrays. A suite of utility routines is available to allocate and release blocks of dynamic memory. These blocks reside in the physically allocated memory region as shown in Figure 7. The actual size of the memory block is determined at execution time. Modules using this feature may be designed to allow *spill logic* which allows operations to be performed on data that exceeds the size of available memory. Dynamic memory management is also used by the database in performing its buffered I/O functions. This represents an extension to the NASTRAN open core concept in that the application programmer is able to manipulate memory blocks rather than being given the total memory available in one block.

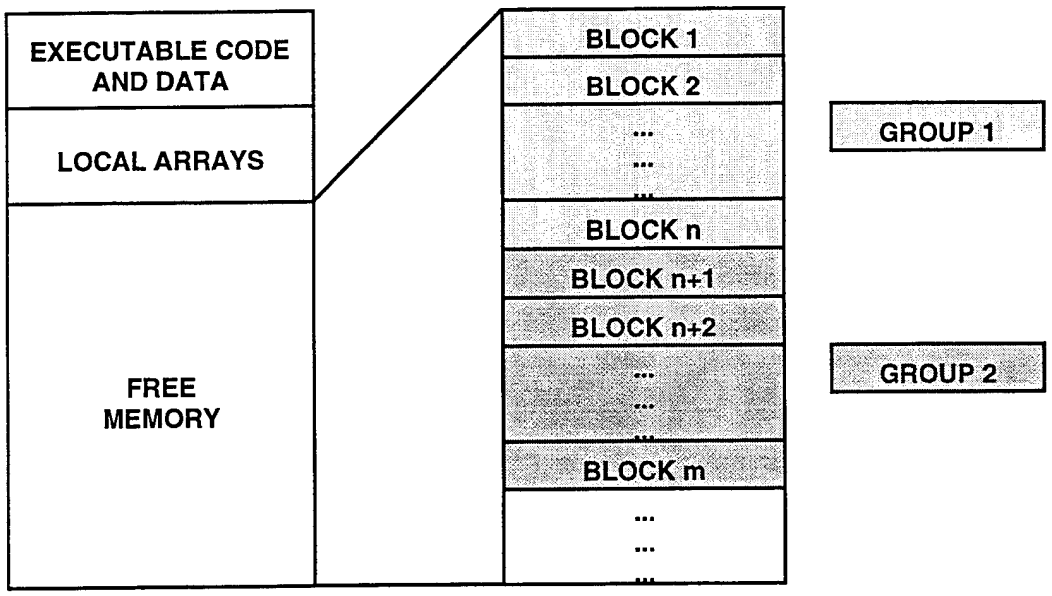


Figure 7. Dynamic Memory

3.4. THE USER INTERFACE

The User Input and Solution Results blocks of Figure 3 represent the user's interface with ASTROS. A very brief discussion is provided of these blocks here since the entire User's Manual is devoted to the documentation of these files.

The User Input is a series of optional packets that are interpreted by the Executive system to direct the design and analysis tasks. The first packet contains "Debug" directives that can be used by a sophisticated user to diagnose problems with the execution. This packet ***is never*** required, but may be useful in diagnosing software problems. The second packet contains the MAPOL sequence which directs the flow of execution. This packet is optional, since the standard MAPOL sequence is available to handle the majority of ASTROS tasks.

A third input packet contains Solution Control directives that select the design and analysis tasks, including the boundary conditions and the required analysis disciplines. This packet also provides output requests that define the majority of the Solution Results outputs. While not required, this packet is almost always needed to direct the procedure.

The fourth input packet is the Function packet, which is also optional. The Function packet is used if the design model includes functional constraints or a functional objective.

The final input packet contains the bulk data which defines the physical and geometric characteristics of the structural system that is to be analyzed and designed. The formats of these data entries are compatible with those used in NASTRAN, to the maximum extent possible. The bulk data packet is almost always required, with the exception of certain "restart" runs where an initial run has completely specified the problem to be analyzed.

The Solution Results that are output are intended to provide the user with the ability to assess the performance of ASTROS on the designated task. Since a multidisciplinary design task could potentially produce an overwhelming amount of output, an effort was made to provide minimal default output. Instead, Solution Control commands provide a means of selecting specific quantities for output. Additional output is available by turning on print requests that are imbedded in calls to functional modules in the MAPOL sequence. This latter type of output requires a modification to the standard MAPOL sequence and is typically of minimal interest to a routine user. Finally, utilities allow the user to print database information to the user's output.

3.5. ENGINEERING MODULES

The engineering modules of Figure 3 are those which perform the specific engineering tasks required in the ASTROS system. The remainder of this report is concerned with describing the algorithms used in these tasks so that this discussion will be limited to what characterizes an engineering module.

The concept of modular programming is essentially one of dividing the overall programming tasks into a number of non-interacting units that can be separately designed and implemented. Input and output data are rigorously defined and control is sequentially passed from one module to another. In the ASTROS system, the Executive System provides this control so that an engineering module can only be accessed through the MAPOL sequence. Modular independence is enforced by requiring that:

- each module establish its own base address in dynamic memory
- database entities required by a module must be opened before their data can be accessed
- all database entities must be closed before the module is exited
- all dynamic memory must be freed before the module is exited

In essence, the requirement of modularity is that all intermodular data communication take place through rigorously defined data formats on the database.

One exception to this module independence in ASTROS is that there is a limited amount of data that are passed through common blocks at the system level. These data include items such as unit number for the read and write files, engineering constants and conversion factors (e.g., pi and the radian to degrees conversion) and system dependent numbers, such as number of lines per page in the output. It would, of course, be possible to independently define these quantities in each module, but this creates other bookkeeping problems. This form of communication is considered part of the executive system since the data are global and the communication is one way. That is, the executive "tells" a module the output logical unit number, never vice versa. Also, this form of communication is never used to pass data between modules.

General utilities perform relatively simple functions that are required repeatedly in any program like ASTROS. Examples are data sort and search routines, CPU timers, data converters and print controllers. A particular reason for identifying and segregating these functions is to avoid duplication of code when two programmers have a similar requirement. Another reason is that a number of these functions are machine dependent so their segregation aids in the installation of ASTROS on a new computer system.

Large matrix utilities are a suite of routines that perform operations on the matrix database entities discussed in Subsection 3.2.1. It is these utilities that permit ASTROS to address problems of essentially unlimited size. Table 5 defines the large matrix utilities available in ASTROS. Since these functions are required repeatedly in a structural analysis task, these utilities can be accessed either directly from the executive system or from the functional modules, as shown in Figure 3. Not all utilities have this feature and those that do require an interface routine between the executive system and the utility.

Table 5. Large Matrix Utilities

UTILITY	FUNCTION
PARTN	$A \rightarrow \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$
MERGE	$A \leftarrow \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$
SDCOMP	$A \rightarrow LDL^T$
FBS	$X = LDL^{-T}B$
DECOMP	$A \rightarrow LU$
GFBS	$X = LU^{-1}B$
MXADD	$C = \alpha A + \beta B$
MPYAD	$D = AB + C$
TRNSPOSE	$B = A^T$
REIG	$[K - \lambda M]\varphi = 0$
CEIG	$\left[p^2 M + pB + K\right]\varphi = 0$

Considering this fact, the distinction between an engineering module and a utility called by the executive is blurred. As an example, the large matrix utility to multiply matrices can be viewed as either an engineering module or a utility. For the purposes of this discussion, it is designated a utility, with the term engineering module reserved for the basic engineering tasks. The distinction being that an engineering module may call a utility through its application interface but may never call another engineering module. The executive system may call both engineering modules and utility modules.

4. MULTIDISCIPLINARY ANALYSIS AND DESIGN

4.1. INTRODUCTION

The ASTROS system was developed to have maximum impact at the preliminary design stage of an aerospace structural design. At this stage, the configuration has been defined and the materials have been selected. The design task is the determination of structural sizes that will provide an optimal structure while satisfying the numerous requirements that multiple disciplines impose on the structure. A key motivation for the development of a single automated structural optimization tool is that such a tool can shorten the design cycle time and provide better structural designs. This is particularly true as composite materials come into widespread use. Balancing conflicting requirements for the strength and stiffness of the structure while exploiting the benefits of anisotropy (e.g., "aeroelastic tailoring") is perhaps an impossible task without assistance from an automated design tool. The use of a single tool can also bring the design task into better focus among design team members, thereby improving the insight into their overall task.

The development of a system to meet these needs is by no means a new endeavor. Concepts of automated structural design have been advanced for over 30 years and a number of software procedures have been developed. Notable among these are the TSO (Reference 4) and FASTOP (Reference 5) procedures that were developed under Air Force sponsorship. NASA has been very active in this area and has sponsored, or performed in-house, many programs that have served to crystallize the methodologies that are applicable in this area (References 6 and 7).

The basic objective in developing the ASTROS system has been to provide a state-of-the-art design tool that integrates existing methodologies into a unified multidisciplinary package. Concepts from TSO and FASTOP were adapted for ASTROS; for example, TSO's capability to simultaneously design to strength, flutter, displacement, and other requirements has been incorporated into ASTROS, as has FASTOP's use of finite element structural analysis.

The distinctive attribute of ASTROS is the scope of conditions it can consider in a design task. Multiple boundary conditions, each permitting a range of analyses (e.g., statics, modes and flutter) can be treated. Also, limits on problem size have been removed for the most part.

The remainder of this section describes the implementation of multidisciplinary analysis and design in ASTROS; first by providing an overview of the design algorithm and then by defining the design task in a mathematical and a physical sense.

4.2. MULTIDISCIPLINARY OPTIMIZATION

A general optimization task may be defined in a mathematical form as:

Find the set of design variables, $\boldsymbol{v}$, which will minimize an objective function

$$\boldsymbol{F}(\boldsymbol{v}) \tag{4-1}$$

subject to constraints:

$$g_j(\boldsymbol{v}) \leq 0.0 \qquad j = 1, \ldots ncon \tag{4-2}$$

$$h_k(\boldsymbol{v}) = 0.0 \qquad k = 1, \ldots ne \tag{4-3}$$

$$v_i^{lower} \leq v_i \leq v_i^{upper} \qquad i = 1, \ldots ndv \tag{4-4}$$

where g specifies the *ncon* inequality constraints and h refers to the *ne* equality constraints. Equation 4-4 specifies upper and lower bounds (side constraints) on each of the design variables. Subsection 4.2 provides the physical interpretations of each of these quantities as they are applied in ASTROS.

Figure 8 presents a schematic diagram of the ASTROS program flow for the design portion of the procedure and contains a number of key concepts that need to be understood in order to appreciate the generality and power of the procedure. The figure indicates that the task is divided into three phases. In the first phase, an analysis of a specified design is performed. As the diagram shows, there can be any number of boundary conditions included in this phase and each boundary condition can contain a number of disciplines. Further, each discipline could contain a number of subcases. As an example, a typical design task could be to analyze the structure for strength at a number of flight conditions (specified by Mach Number, altitude and load factor) and also to evaluate the flutter behavior at another set of flight conditions for both symmetric and antisymmetric response. It should be clear that each of these conditions could contain a response that is critical in determining the design and that all critical conditions must be considered simultaneously to achieve an overall best design. The inability of previous automated design procedure to perform this simultaneous analysis has been seen as one of their primary weaknesses by potential users.

As Figure 8 shows, each of the subcases generates constraints that quantify the response of the design relative to prescribed limits. In the second phase, the sensitivities of these constraints to changes in the design variables are calculated. Note that this discussion of the sensitivity and the optimization phases pertains only to the mathematical programming option for design. Subsection 13.2 discusses the alternative Fully Stressed Design option.

Because of the potentially large number of constraints, a screening process takes place to select the constraints that can be expected to play a role in the redesign (see Subsection 13.1). Two important points to be made for the present discussion are that (1) the sensitivity calculations require a looping through the same boundary condition, discipline and subcase hierarchy that was required in the analysis phase and (2) it would be inefficient to calculate these sensitivities "on the fly" during the analysis phase, since only a small percentage of the constraints require sensitivities and the identity of the "active" constraints cannot be determined until all the constraints are known.

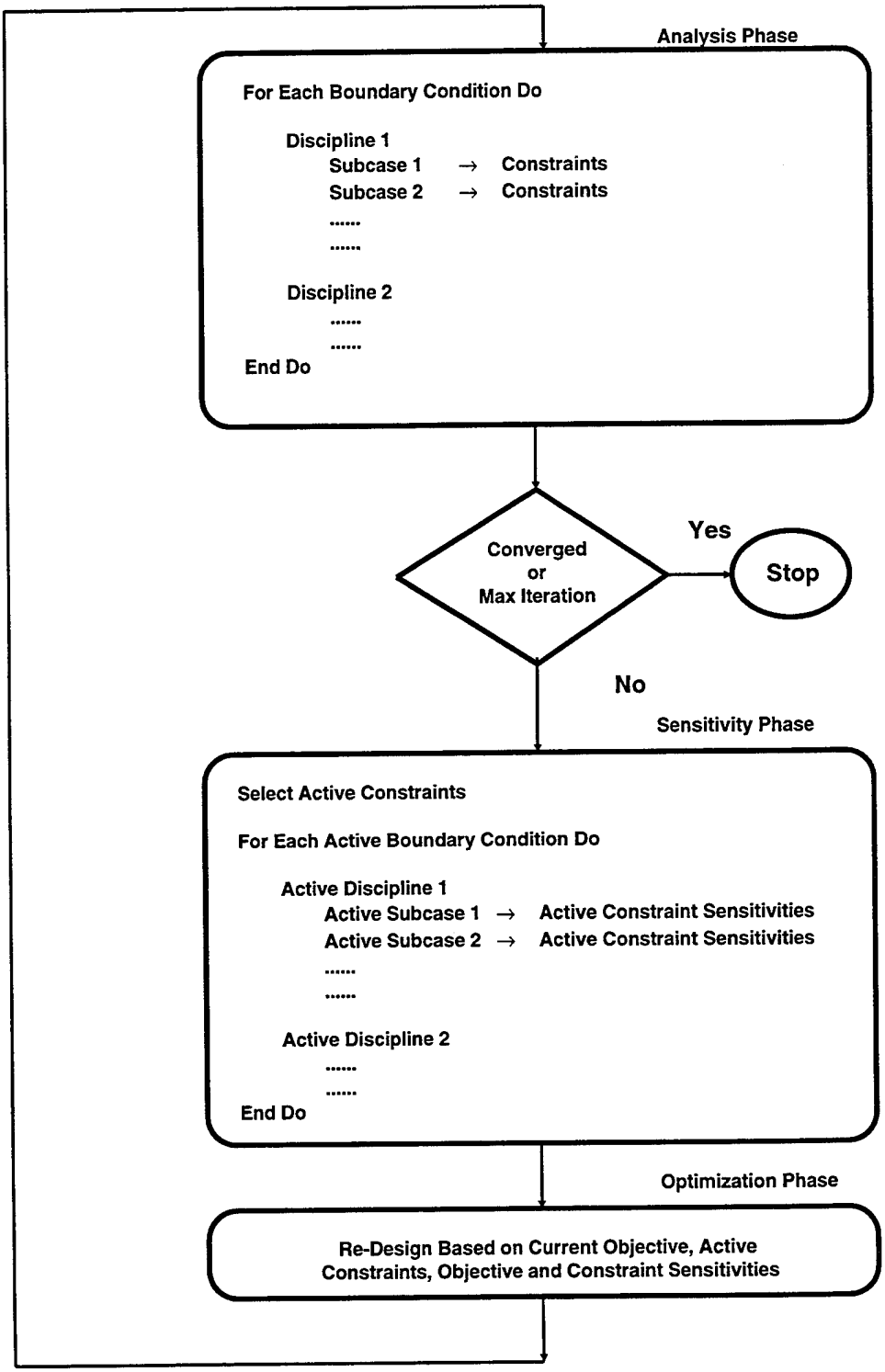


Figure 8. Multidisciplinary Optimization

In the optimization phase, the information on the objective and the active constraints is assimilated into a redesign algorithm so as to meet the requirements of Equations 4-1 through 4-4. Subsection 13.1 describes how this information is utilized to the maximum practical degree so that the iterations through the computationally expensive analysis and sensitivity phases are kept to a minimum. As a final point on Figure 8, the convergence test for program termination entails an evaluation of whether the redesign is making progress in meeting the requirements or if the maximum specified number of iterations have been made.

Equations 4-1 through 4-4 are general in the sense that they apply to any optimization task. The following subsections describe the meaning of each of these terms in the equations in ASTROS.

4.3. Objective Function

The objective to be minimized may be structural weight, a synthetic function of the structural responses, or characteristics of the finite element model. There are no equality constraints in ASTROS; therefore, there is no need to further consider Equation 4-3. The remaining terms require substantially more definition.

Synthetic functions are available in ASTROS to play the role of either the objective function or a constraint function. Synthetic functions are formed as mathematical combinations of analytical responses, such as stresses and flutter roots, or model characteristics, such as weight and thickness. Section 4.6 presents the manner in which synthetic functions are introduced into the optimization process.

4.4. Design Variables

ASTROS defines design variables at two levels: (1) Physical (or local) variables, and (2) Global Variables. The basic rationale for having these two levels is to reduce the number of design variables to a number that is tractable in a mathematical programming context. As will be discussed, a further motivating factor is that it provides the user with a means of imposing constraints on the design task that are desirable due to manufacturing or other considerations. This is not a new concept; for example Reference 6 provides a discussion and review of techniques for reducing the number of design variables.

4.4.1. Local Variables

These variables are properties of the finite elements used in ASTROS. Table 6 lists the finite element types that can be designed and their associated design variables.

Most of the physical design variables in ASTROS share a common property: their associated element mass and stiffness matrices are a linear function of the design variable. This fact is exploited by computing the invariant portions of these matrices only once in the preface portion of the procedure. This portion is then multiplied by the current value of the design variable during the assembly of the global stiffness and mass matrices.

There are a number of exceptions to this linear behavior. The nonlinear behaviors fall into one of three broad categories: (1) a design invariant property coupled with linear behavior, (2) the factorable

Table 6. Physical Design Variables

ELEMENT	DESIGN VARIABLE
CROD	Area
CSHEAR	Thickness
CQDMEM	Thickness(es)
CTRMEM	Thickness(es)
CQUAD4	Membrane, bending, or composite layer thickness(es)
CTRIA3	Membrane, bending, or composite layer thickness(es)
CBAR	Area, Cross-sectional parameters
CONM2	Mass
CELAS1,2	Stiffness
CMASS1,2	Mass

BAR element, and (3) fully nonlinear relationships. Each of these categories is treated differently as descibed in the following.

1. Whenever an element has a set of properties that is design invariant and another set that is linear, ASTROS creates a pseudo-element that contains the design invariant properties while the remaining properties are factored as a linear design variable. In this manner, the benefits of linearity can still be exploited. Examples of this category include the nonstructural mass of any element, the torsional stiffness of the ROD element, and the inertia terms of a CONM2 element.
2. When a general one-dimensional bending element is designed, there are three possible means to define the design variables and their relationship to the element properties: (1) use cross-sectional geomtric parameters as design variables and derive the area, moments of inetia, and stress recovery points from these parameters; (2) independently design the general properties; or (3) use a single design variable and prescribe a relationship among the elemental properties and this variable.

   The third approach is used to allow linear factorization of the BAR element while retaining some reasonable generality. In this approach, the inertial properties (I_1 and I_2) are prescribed, exponential functions of the area (A):

$$I_1 = r_1 A^{\alpha}$$
$$I_2 = r_2 A^{\alpha} \qquad (4\text{-}5)$$

   Where r_1, r_2 and α are user defined quantities, and I_{12} , pin flags and other specialized BAR options are disallowed. Further, the stress recovery points are assumed to be design invariant. This artificial construct maintains linear factorization in A and A^{α} and does allow adequate

modeling of common BAR geometries ($\alpha = 1$ corresponds to a thin walled beam while $\alpha = 3$ corresponds to a solid beam of constant width and varying depth).

The second method suffers from serious drawbacks for beam design by treating variables which are dependent in an independent manner. Therefore, it is entirely possible that the resultant optimal BAR is one that cannot be manufactured. To remove this possibility, ASTROS designs BAR elements using the first method in which the design variables are the actual cross-sectional geometric parameters. This is a in a fully nonlinear approach manner as described next.

3. Whenever there is no factorable relationship between the stiffness or mass matrices and the independent variables, ASTROS uses finite difference techniques to compute the elemental matrix sensitivities and recomputes the exact elemental matrices at each new design point. Examples of this approach include BAR cross-sectional design variables (CBAR,PBAR1) and plate bending elements. For plate bending elements defined by shell properties (PSHELL), the relationship between the properties and the independent variable, t, are merely scaled by the design variable. In other words, for sandwich composites defined by PSHELL data, the design variable represents the smeared thickness variable. For isotropic plates, this relationship is exact. For nonisotropic plates, use of a composite layup definition allows ASTROS to retain the correct relationship among the elemental properties and the independent variables.

As a final point, references in Table 6 to thicknesses for the membrane elements refer to the fact that the thickness of each ply direction for a composite element can be treated as a separate design variable. This emulates the TSO (Reference 4) and FASTOP (Reference 5) treatments of composite materials.

4.4.2. Global Variables

These variables are the ones that are directly involved in the design process. The local variables are linked to the global values through a matrix relationship of the form:

$$\boldsymbol{t} = \boldsymbol{P}\boldsymbol{v} \tag{4-6}$$

where $\boldsymbol{t}$ is a vector of *nloc* local variables, $\boldsymbol{v}$ is vector of *ndv* global variables and $\boldsymbol{P}$ is the linking matrix of dimension *nloc* by *ndv*. Three linking options are provided in ASTROS.

4.4.2.1. Unique Linking

In this case, the global variables are the same as the local variables and there is a single nonzero term in the corresponding row of the linking matrix and its value is the initial local property value.

4.4.2.2. Physical Linking

One global variable uniquely specifies a number of local variables. This option is used to permit the simultaneous variation of finite elements over a region of the structure, the rationale being that there is no inherent reason why each finite element should be independently designed. There may be manufacturing reasons why this linking should occur or it may be that the designer knows that uniform properties in certain areas of the structures are adequate. The corresponding rows of the $\boldsymbol{P}$ matrix for the local variables have a single nonzero term corresponding to the initial local property value.

4.4.2.3. Shape Function Linking

A local variable is the weighted sum of several global variables. In this case, the global variable controls the magnitude of a shape function that applies over a region of the structure. The shape function concept is best illustrated by reference to TSO's representation of the skin thickness as being the weighted sum of polynomials in the non-dimensional coordinates ξ and η of the trapezoidal wing box:

$$t(\xi,\eta) = \sum_{i=1}^{3} \sum_{j=1}^{3} a_{ij}\, \xi^{i-1}\, \eta^{j-1} \tag{4-7}$$

where the a_{ij} are the design variables. ASTROS has expanded this capability by allowing the user to define any shape function over any part of the structure. For this third case, a row of the $\boldsymbol{P}$ matrix can have any number of nonzero terms, and they can be either positive or negative. These factors are applied to a unit local property value in computing the local variable.

4.4.2.4. Linking Example

Further perspective on these aspects can be obtained by referring to the simple model shown in Figure 9, which is the Intermediate Complexity Wing (ICW) used by Grumman in the development of the FASTOP procedure. The model has 62 quadrilateral membrane elements that represent the upper and lower skin surfaces. Each of these elements contains four layers of composite material. A number of different linking concepts can be studied using the ASTROS procedure. In one, all the elements between

NO. OF NODES	NO. OF ELEMENTS		NO. OF DOF	
88	39	RODS	294	RODS
	55	SHEAR PANELS	234	SHEAR PANELS
	62	QUADRILATERAL MEMBRANE	528	TOTAL
	2	TRIANGULAR MEMBRANE		
	158	TOTAL		

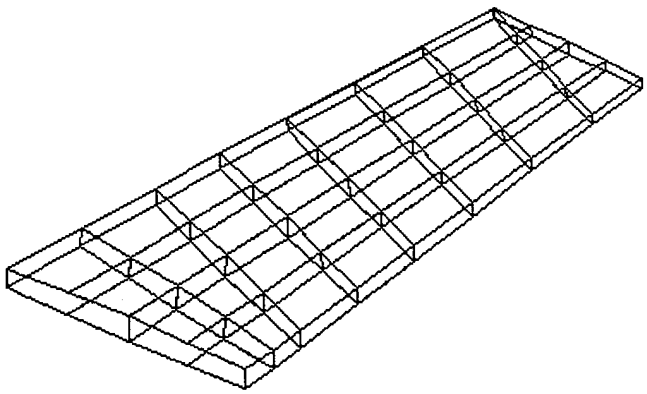


Figure 9. Intermediate Complexity Wing Structure

two ribs could be linked to give the same thickness, with different thicknesses allowed on the two surfaces. This would result in 2 (surfaces) x 4 (layers) x 8 (bays) = 64 global design variables. Alternatively, the user could allow the thickness to vary linearly in the spanwise direction while holding it constant in the chordwise direction. This could provide a reasonable design that is also attractive from a manufacturing standpoint. There would then be one global design variable for each surface that specifies the level of a uniform distribution of the thickness while a second variable provides the linear taper. This is equivalent to designating the a_{11} and a_{12} components of Equation 4-7 as design variables while setting the remainder of the components to zero. This results in 2 (surfaces) x 4 (layers) x 2 (shapes) = 16 global design variables.

It is recognized that the flexibility provided by these three options also places a burden on the users in term of defining the design variables. Subsection 3.1 of the Applications manual discusses the preparation of the bulk data inputs for these three options in some detail. Subsection 4.7 and 4.8 of the same manual contain results from applying a variety of linking options to the ICW of Figure 9.

4.5. Constraints

Constraints in ASTROS are of three basic types: constraints on directly computed response quantities, as given by Equation 4-2, constraints on linear or nonlinear functions of computed responses, also represented symbolically by Equation 4-2 and side constraints on the design variables, as given by Equation 4-4. The design variable options described in the previous subsection complicate the definition of side constraints so that these constraints are included here in the discussion of thickness constraints. The response constraints are divided into those that represent strength constraints and those that represent stiffness constraints. The constraints are introduced in this subsection, with more detailed descriptions deferred until the discussion of their associated disciplines in Sections 6, 7, 9, and 10.

4.5.1. Strength Constraints

Four strength constraints are provided in ASTROS as described in the following sections.

4.5.1.1. Von Mises Stress Constraint

The Von Mises stress constraint in ASTROS is more properly viewed as the Tsai-Hill failure criteria, which is the anisotropic extension of the Hencky-Von Mises criterion. This constraint on element stress is written in the format of Equation 4-2 as:

$$g = \left[\left(\frac{\sigma_x}{S_x}\right)^2 + \left(\frac{\sigma_y}{S_y}\right)^2 - \frac{\sigma_x \sigma_y}{S_x S_y} + \left(\frac{\tau_{xy}}{F_s}\right)^2 \right]^{\frac{1}{2}} - 1.0 \qquad (4\text{-}8)$$

where x and y are the normal stresses in the element coordinate system and xy is the corresponding transverse shear stress. F_s is a user defined limit for the shear stress while S_x and S_y are allowables for σ_x and σ_y that are derived from S_t and S_c, the tensile and compressive allowables, based on the sign of σ_x and σ_y. For the special case of isotropic materials with $S_t = S_c = S$, the equivalent Hencky-Von Mises

criterion is obtained by using $F_s = \frac{1}{\sqrt{3}} S$. The tension and compression limits need not be the same so that, in evaluating Equation 4-8, the sign of the normal stresses must be known before the appropriate divisor can be selected.

4.5.1.2. Tsai-Wu Stress Constraint

This constraint on element stress is based on the Tsai-Wu failure criterion (Reference 8) which states that a material will fail when

$$F_{ij}\,\sigma_i\,\sigma_j + F_i\,\sigma_i = 1.0 \tag{4-9}$$

For the two-dimensional elements of ASTROS, this becomes:

$$F_{11}\sigma_1{}^2 + 2F_{12}\sigma_1\sigma_2 + F_{22}\sigma_2{}^2 + F_1\sigma_1 + F_2\sigma_2 + F_{66}\tau^2{}_{12} = 1.0 \tag{4-10}$$

where symmetry considerations dictate that the F_{16}, F_{26} and F_6 terms are zero. The remaining coefficient terms are:

$$\begin{aligned} F_{11} &= \frac{1}{x_t x_c} \\ F_{22} &= \frac{1}{y_t y_c} \\ F_1 &= \frac{1}{x_t} - \frac{1}{x_c} \\ F_2 &= \frac{1}{y_t} - \frac{1}{y_c} \\ F_{66} &= \frac{1}{s^2} \end{aligned} \tag{4-11}$$

where x, y and s are allowables in the longitudinal, transverse and shear directions for a fiber and the t and c subscripts refer to tension and compression. The F_{12} term is not defined analytically, instead it must be provided by experiment for each material. There are, however, certain limits to the value that F_{12} can take.

The Tsai-Wu criterion is utilized in ASTROS by determining the strength ratios, R, that the stress state must be multiplied by to exactly satisfy Equation 4-10. This factor is determined by solving the quadratic equation:

$$aR^2 + bR - 1.0 = 0.0 \tag{4-12}$$

where:

$$\begin{aligned} a &= F_{11}\sigma_1^2 + F_{22}\sigma_2^2 + F_{66}\tau^2_{12} + 2F_{12}\sigma_1\sigma_2 \\ b &= F_1\sigma_1 + F_2\sigma_2 \end{aligned} \tag{4-13}$$

By examination of Equation 4-12 we can obtain some idea about the size of F_{12}. From Equation 4-12 we obtain

$$R = \frac{-b \pm \sqrt{b^2 + 4a}}{2a} \tag{4-14}$$

where b, expanded, is

$$b = \left(\frac{1}{x_t} - \frac{1}{x_c}\right)\sigma_1 + \left(\frac{1}{y_t} - \frac{1}{y_c}\right)\sigma_2 \tag{4-15}$$

in which all the allowable values $\left(x_t, x_c, y_t, y_c\right)$ are positive. If we assume that the tension and compression allowables are equal, b is identically zero. Therefore, for R to be defined as a real number, a must be positive. This leads to a set of limits on F_{12} to ensure under all circumstances that R is defined:

$$-\frac{1}{\sqrt{x_t x_c y_t y_c}} \leq F_{12} \leq \frac{1}{\sqrt{x_t x_c y_t y_c}} \tag{4-16}$$

This expression, however, is overly restrictive in the case where the tension and compression allowables are different ($b \neq 0$). In that case the more general requirement that $b^2 + 4a$ be positive applies instead and F_{12} cannot be bounded without knowing the stress state. In ASTROS, the condition of Equation 4-16 is imposed during the input processing and results in a warning if not satisfied. Subsequently the general requirement is, of course, imposed with fatal errors issued if the R value is not real.

The constraint is formed as

$$g = \frac{1.0}{R} - 1.0 \tag{4-17}$$

4.5.1.3. Principal Strain Constraint

The implementation of a strain constraint in ASTROS is based on the two principal strains in a two-dimensional element:

$$\varepsilon_x = \frac{1}{2}\left[\varepsilon_1 + \varepsilon_2 + \left[\left(\varepsilon_1 - \varepsilon_2\right)^2 + \varepsilon_{12}^2\right]^{1/2}\right]$$

$$\varepsilon_y = \frac{1}{2}\left[\varepsilon_1 + \varepsilon_2 - \left[\left(\varepsilon_1 - \varepsilon_2\right)^2 + \varepsilon_{12}^2\right]^{1/2}\right] \tag{4-18}$$

Two constraints are computed per element based on the strains of Equation 4-18, with the evaluation dependent on whether the user has specified a single strain limit or if separate tension and compression (ε_T and ε_c) allowables are specified.

If $\varepsilon_c = 0$, the constraints are calculated using

$$g_1 = \frac{\varepsilon_x}{\varepsilon_{all}} - 1.0$$
$$g_2 = \frac{\varepsilon_y}{\varepsilon_{all}} - 1.0 \qquad (4\text{-}19)$$

If ε_c is nonzero, similar formulas are used, with the selection of ε_{all} based on the sign of the computed strains. For example, if ε_y is negative, then

$$g_2 = \frac{|\varepsilon_y|}{|\varepsilon_c|} - 1.0 \qquad (4\text{-}20)$$

4.5.1.4. Fiber/Transverse Strain Constraint

The implementation of a fiber and transverse strain constraint in ASTROS is based on the strains along the material axis and transverse to it. Each layer of a composite element can have different values of fiber and transverse strain, even in the absence of bending effects.

One or two constraints are computed per element per layer based on the strains computed in a coordinate system aligned with the fiber direction of the layer. Constraints are computed if the appropriate allowable has been specified. Computation of the constraint is identical to Equation 4-19, except that the strain components are those in the fiber/transverse axes. Again, if separate tension and compression (ε_{f_T}, ε_{t_T} and ε_{f_c},ε_{t_c}) allowables are specified, the appropriate allowable is used.

4.5.2. Stiffness Constraints

A number of the constraints imposed in ASTROS can be thought of as placing limits on the structural stiffness. Although inertia properties will play a role in some of these constraints, it is still a convenient distinction, with displacement, frequency, flutter and static aeroelastic conditions the available stiffness constraints in ASTROS.

4.5.2.1. Displacement Constraints

Displacement constraints are either upper bound:

$$\sum_{j=1}^{ndisp} A_{ij} u_j \leq \delta_{i_{all}} \qquad (4\text{-}21)$$

or lower bound:

$$\sum_{j=1}^{ndisp} A_{ij} u_j \geq \delta_{i_{all}} \tag{4-22}$$

where the A_{ij} are user specified weighting factors on structural displacement and δ_i is the user specified limit. Note that the summation permits the specification of limits on the shape of a deformation. For example, the twist of a wing tip could be limited by differencing the displacements at the leading and trailing edges of the structural torque box:

$$\frac{(w_{LE} - w_{TE})}{c_{TIP}} \leq 0.04 \text{ radians} \tag{4-23}$$

where c_{TIP} is the chord distance between two displacements.

4.5.2.2. Frequency Constraints

Limits on the natural frequencies of the structure can be specified as

$$f_{low} \leq f_i \leq f_{high} \tag{4-24}$$

where f_i is the computed value of the i^{th} natural frequency and f_{low} and f_{high} are user specified limits on this frequency. Note that formulation permits the specification that a frequency be within a certain band, but it does not allow the exclusion of a frequency from a range:

$$f_i < f_{low} \text{ or } f_i > f_{high} \text{ for } f_{low} < f_{high} \tag{4-25}$$

The difficulty is that ASTROS does not permit the "or" type of specification. Furthermore, if the frequency did lie in the excluded zone, it is not easy to specify a redesign algorithm that could determine whether it is better to drive the frequency up or down.

To improve the accuracy of the approximations used in the optimization process, ASTROS imposes the frequency constraint on the equivalent eigenvalue, i.e:

$$f_i \leq f_{high} \Rightarrow g = 1 - \frac{4\pi^2 f_{high}^2}{\lambda_i} \quad \text{and} \quad f_{low} \leq f_i \Rightarrow g = \frac{4\pi^2 f_{low}^2}{\lambda_i} - 1$$

4.5.2.3. Flutter Constraint

The flutter constraint in ASTROS is formulated in terms of satisfying requirements on the modal damping values at a series of user specified velocities:

$$\gamma_{ij} \leq \gamma_{j\,REQ} \qquad j = 1, 2 \ldots nvel \tag{4-26}$$

where $\gamma_{j\,REQ}$ is the required level of damping at the j^{th} velocity and γ_{ij} is the computed damping level for the i^{th} branch at the j^{th} velocity. A further discussion of this constraint is given in Subsection 10.3, following the development of the flutter equations. A point to be made here is that the constraint formulation of Equation 4-26 does not require the determination of the flutter speed.

4.5.2.4. Lift Effectiveness Constraint

The lift effectiveness constraint places bounds on the ratio of the flexible to rigid lift curve slope of the aircraft:

$$\varepsilon_{min} \leq \frac{C_{L_{\alpha_f}}}{C_{L_{\alpha_r}}} \leq \varepsilon_{max} \tag{4-27}$$

where $C_{L_{\alpha_f}}$ is the flexible lift curve slope and includes the effects of aeroelastic deformation and inertia relief. $C_{L_{\alpha_r}}$ is the lift curve slope for the rigid aircraft. This constraint gives the user a direct and physically meaningful way of controlling the amount of flexibility in the structure.

4.5.2.5. Aileron Effectiveness Constraint

Roll performance requirements frequently drive the design of aircraft wing structures. This factor has been recognized in ASTROS by the incorporation of an aileron effectiveness constraint. Aileron effectiveness, following terminology used in Reference 9 can be defined as the ratio of roll due to aileron deflection over roll due to roll rate:

$$\varepsilon_{eff} = -\frac{\left(C_{l_{\delta_a}}\right)^f}{\left(C_{l_{\frac{pb}{2V}}}\right)^f} \tag{4-28}$$

where

C_l = Rolling moment about the aircraft center line

δ_a = Aileron deflection

$\frac{pb}{2V}$ = Roll rate nondimensionalized by wing span and aircraft velocity

f = Flexibility effects are included in the derivatives

The effectiveness parameter can be thought of as a measure of the steady roll rate achievable for a unit value of aileron deflection. In a manner similar to the lift effectiveness, the user can specify that the aileron effectiveness be within a specified range:

$$\varepsilon_{min} \leq \varepsilon_{eff} \leq \varepsilon_{max} \tag{4-29}$$

An intriguing application of this constraint is its application to specify a reversed aileron. In this case the effectiveness limits would be negative and active controls would typically be necessary to augment the aircraft performance.

4.5.2.6. Flexible Stability Derivative Constraint

A more general constraint on the stability derivatives is also available that allows the flexible derivative in any axis for any trim parameter to be constrained using:

$$\left[\frac{\partial C_F}{\partial \delta_{trim}}\right]_{lower} \leq \frac{\partial C_F}{\partial \delta_{trim}} \leq \left[\frac{\partial C_F}{\partial \delta_{trim}}\right]_{upper} \tag{4-30}$$

4.5.2.7. Trim Parameter Constraint

A final constraint that can be applied in the aeroelastic domain is one that allows limits on the values of the computed trim parameters, e.g., angle-of-attack or control surface deflection. These constraints have very limited application, but may be useful in controlling the optimization process when it moves the design to a point in which the linear aerodynamics are no longer valid. For example, if the "optimal solution" has a 60 degree aileron deflection, the aerodynamics of ASTROS are no longer valid. It may be possible that a constraint on aileron deflection will yield a more useful design.

The constraints that can be applied are of the form:

$$\delta_{trim} \leq \delta_{trim_{Req}} \quad \text{or} \quad \delta_{trim} \geq \delta_{trim_{Req}} \tag{4-31}$$

and can be applied to any FREE trim parameter in any static aeroelastic analysis.

4.5.3. Buckling Optimization — The Unstiffened Plate

A capability has been implemented which allows a panel buckling constraint to be specified in ASTROS. The full capability allows modeling of stiffened panels by considering the plate bending behavior and the stiffener behavior independently but simultaneously. Thus there is an unstiffened panel buckling constraint driven by plate bending elements and an Euler buckling constraint driven by one-dimensional bending elements. This section outlines the theoretical development for the simply supported option of the plate buckling constraint and describes the manner in which it is used.

Consider the panel shown in Figure 10. The panel, which is simply supported on all four sides, has dimensions a and b. The running loads are shown as N_x, N_y, and S. Laminate bending stiffness is represented as D_{ij}: $i = 1,...,6; j = 1,...6$. The governing differential equation for the plate is:

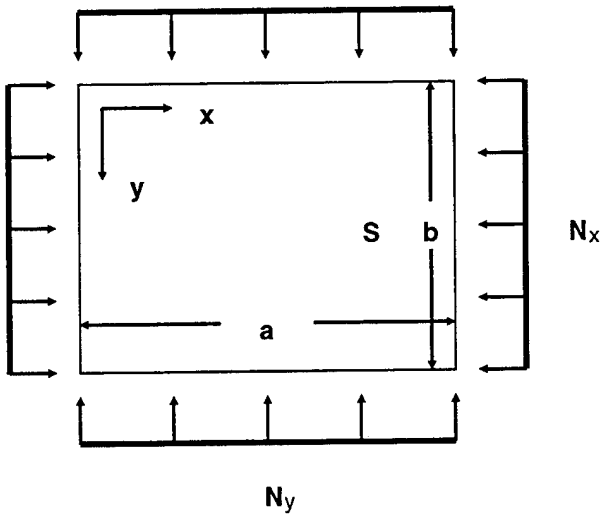


Figure 10. Buckling Panel

$$D_{11}\frac{\partial^4 w}{\partial x^4} + 4D_{16}\frac{\partial^4 w}{\partial x^3\,\partial y} + 2(D_{12}+2D_{66})\frac{\partial^4 w}{\partial x^2\partial y^2} + 4D_{26}\frac{\partial^4 w}{\partial x\,\partial y^3} + D_{22}\frac{\partial^4 w}{\partial y^4}$$
$$= N_x\frac{\partial^2 w}{\partial x^2} + N_y\frac{\partial^2 w}{\partial y^2} + 2S\frac{\partial^2 w}{\partial x\,\partial y} \qquad (4\text{-}32)$$

with boundary conditions:

at $x = 0, a$

$$w = M_x = -D_{11}\frac{\partial^2 w}{\partial x^2} - 2D_{16}\frac{\partial^2 w}{\partial x\,\partial y} - D_{12}\frac{\partial^2 w}{\partial y^2} = 0 \qquad (4\text{-}33)$$

at $y = 0, b$

$$w = M_y = -D_{12}\frac{\partial^2 w}{\partial x^2} - 2D_{26}\frac{\partial^2 w}{\partial x\,\partial y} - D_{22}\frac{\partial^2 w}{\partial y^2} = 0 \qquad (4\text{-}34)$$

In many circumstances, Galerkin's method can be used to solve such equations by selecting a series of functions, each of which satisfy the boundary conditions, and then determining the coefficients in the series by requiring that the solution be orthogonal to each term in the series. Unfortunately, there is no such series which satisfies these boundary conditions.

As a result, boundary integrals must be included in the formulation of the Galerkin equation. The double sine series:

$$w = \sum_{m=1}^{M} \sum_{n=1}^{N} A_{mn} \sin\frac{m\pi}{a}x \sin\frac{n\pi}{b}y \tag{4-35}$$

satisfies the geometrical condition $w = 0$ on the boundary. Noting that on the boundary:

$$\frac{\partial^2 w}{\partial x^2} = \frac{\partial^2 w}{\partial y^2} = 0 \tag{4-36}$$

for the series in (4-35), the Galerkin equation may be written as:

$$\begin{aligned}
&\int_o^b \int_o^a \left[D_{11}\frac{\partial^4 w}{\partial x^4} + 4D_{16}\frac{\partial^4 w}{\partial x^3 \partial y} + 2(D_{12}+2D_{66})\frac{\partial^4 w}{\partial x^2 \partial y^2} + 4D_{26}\frac{\partial^4 w}{\partial x \partial y^3} \right. \\
&\left. + D_{22}\frac{\partial^4 w}{\partial y^4} - N_x\frac{\partial^2 w}{\partial x^2} - N_y\frac{\partial^2 w}{\partial y^2} - 2S\frac{\partial^2 w}{\partial x \partial y} \right] \sin\frac{m\pi x}{a} \sin\frac{n\pi y}{b}\, dx\, dy \\
&- 2D_{26}\int_o^a \left[-1^n \left(\frac{\partial^2 w}{\partial x \partial y}\right)_{y=b} - \left(\frac{\partial^2 w}{\partial x \partial y}\right)_{y=o} \right] \frac{n\pi}{b} \sin\frac{m\pi x}{a}\, dx \\
&- 2D_{16}\int_o^b \left[-1^m \left(\frac{\partial^2 w}{\partial x \partial y}\right)_{x=a} - \left(\frac{\partial^2 w}{\partial x \partial y}\right)_{x=o} \right] \frac{m\pi}{a} \sin\frac{n\pi y}{b}\, dy \\
&= 0 \quad \begin{cases} m = 1,2,\ldots,M \\ n = 1,2,\ldots,N \end{cases}
\end{aligned} \tag{4-37}$$

Substituting (4-35) into (4-37) and integrating yields the algebraic equations:

$$\begin{aligned}
&\pi^4 \left[D_{11}m^4 + 2(D_{12}+2D_{66})m^2 n^2 R^2 + D_{22}n^4 R^4 - N_x\frac{m^2 a^2}{\pi^2} - N_y\frac{n^2 b^2}{\pi^2}R^4 \right] A_{mn} \\
&- 32\, m\, n\, R^2 \pi^2 \sum_{i=1}^{M}\sum_{j=1}^{N} M_{ij} \left[(m^2+i^2)D_{16} + (n^2+j^2)D_{26} - \frac{S}{R}\frac{a^2}{\pi^2} \right] A_{ij} = 0
\end{aligned} \tag{4-38}$$

where:

$$\begin{aligned}
M_{ij} &= \frac{ij}{(m^2-i^2)(n^2-j^2)} \quad \begin{cases} m \pm i = odd \\ n \pm j = odd \end{cases} \\
&= 0 \qquad\qquad\quad \begin{cases} i = m, m \pm i = even \\ j = n, n \pm j = even \end{cases}
\end{aligned} \tag{4-39}$$

and $R = a/b$. Equation (4-38) results in a set of $M \times N$ homogeneous equations. To obtain other than the trivial solution, $A_{mn} \equiv 0$, the determinant of the coefficient matrix must vanish. This introduces the buckling eigenvalue problem that is solved. Before moving to the solution for the buckling eigenvalue, reconsider (4-38). Equation (4-38) may be rewritten as:

$$p^k(m,n)A_{mn} + \sum_{i=1}^{M}\sum_{j=1}^{N} q_{ij}^k(m,n)A_{ij} = \lambda\left[p^q(m,n)A_{mn} + \sum_{i=1}^{M}\sum_{j=1}^{N} q_{ij}^q(m,n)A_{ij}\right] \quad (4\text{-}40)$$

where:

$$\begin{aligned} p^k(m,n) &= \pi^4\left[D_{11}m^4 + 2(D_{12}+2D_{66})m^2n^2R^2 + D_{22}n^4R^4\right] \\ q_{ij}^k(m,n) &= -32\,m\,n\,R^2\pi^2\,M_{ij}\left[(m^2+i^2)D_{16} + (n^2+j^2)D_{26}\right] \\ p^q(m,n) &= N_x m^2 a^2 \pi^2 + N_y n^2 b^2 \pi^2 R^4 \\ q_{ij}^q(m,n) &= -32\,m\,n\,R\;M_{ij}\,S\,a^2 \end{aligned} \quad (4\text{-}41)$$

Alternatively, (4-40) may be expressed in matrix form as:

$$\boldsymbol{K}\Phi = \lambda \boldsymbol{K}_G \Phi \quad (4\text{-}42)$$

where:

$$\Phi^T = \begin{matrix}\{A_{11},A_{12},\ldots,A_{1N},A_{21},A_{22},\ldots,A_{2N},A_{M1},A_{M2},\ldots,A_{MN}\} \\ \{\varphi_1,\varphi_2,\ldots,\varphi_{M\times N}\}\end{matrix}$$

and the subscript of $A_{ij} \rightarrow \varphi_k$ satisfies:

$$k = (i-1)N + j;\;\; i = 1,\ldots,M;\; j = 1,\ldots,N \quad (4\text{-}43)$$

Equation (4-40) represents the eigenvalue problem which must be solved to obtain the multiplier λ on the panel running loads that result in buckling. There are $M \times N$ eigenvalues that can be extracted and the lowest in absolute value represents the critical buckling mode. In the ASTROS implementation, only the lowest eigenvalue for a given panel will be constrained.

In order to obtain a solution to (4-40) for the buckling eigenvalues, the running loads N_x, N_y, and S and the values of D_{ij}, M, N, a and b must be obtained. The running loads and material properties are derived from a finite element selected by the user. These data are then applied to a pseudo-panel whose dimensions are supplied by the user. M and N are selected during the analysis to provide a reasonable combination of numerical accuracy and computational cost.

Given the critical eigenvalue from (4-40), one can relate the critical running load to the eigenvalue:

$$N_c = (N_{cx}^2 + N_{cy}^2 + S_c^2)^{1/2} = \lambda (N_x^2 + N_y^2 + S^2)^{1/2} = \lambda N \tag{4-44}$$

then the constraints that can be imposed are that

$$g = \left[\frac{\lambda_{all}}{\lambda}\right]^{1/3} - 1.0 < 0.0 \quad \text{(lower bound)} \tag{4-45}$$

and

$$g = 1.0 - \left[\frac{\lambda_{all}}{\lambda}\right]^{1/3} < 0.0 \text{ (upper bound)} \tag{4-46}$$

and sensitivity analysis can be performed under the assumption that the running load at the current eigenvalue is a constant. The cube root is used to help improve the linear approximations used during the optimization.

4.5.3.1. Sensitivity Analysis

Constraint sensitivities are computed in a straight-forward manner. Taking the derivative of the upper bound constraint as an example:

$$\frac{\partial g}{\partial v} = \frac{\lambda_{all}}{3\lambda^2}\left(\frac{\lambda_{all}}{\lambda}\right)^{-2/3}\frac{\partial \lambda}{\partial v} = \frac{(1-g)}{3\lambda}\frac{\partial \lambda}{\partial v} \tag{4-47}$$

$$\frac{\partial \lambda}{\partial v} = \frac{\partial \lambda}{\partial D_{ij}}\frac{\partial D_{ij}}{\partial t_l}\frac{\partial t_l}{\partial v} \tag{4-48}$$

where $\frac{\partial t_l}{\partial v}$ comes from the design variable linking matrix $\boldsymbol{P}$. To obtain $\frac{\partial \lambda}{\partial D_{ij}}$, (4-42) is differentiated analytically with the assumption that N is invariant for the current design step:

$$\frac{\partial \lambda}{\partial D_{ij}} = \frac{\Phi^t \left[\frac{\partial \boldsymbol{K}}{\partial D_{ij}}\right] \Phi}{\Phi^t \boldsymbol{K}_G \Phi} \tag{4-49}$$

where the term $\frac{\partial \boldsymbol{K}}{\partial D_{ij}}$ is either obtained analytically from (4-41) or by using finite difference methods. The final term, $\frac{\partial D_{ij}}{\partial t_l}$, which will accommodate both composite and non-composite materials, will always be

computed by central difference methods, since it is relatively inexpensive and will only be computed for active buckling constraints.

4.5.4. Buckling Optimization — The Column

The derivation of the buckling equations for columns is straight-forward. It is based on the simple Euler formula for allowable compressive force for thin walled tubular members:

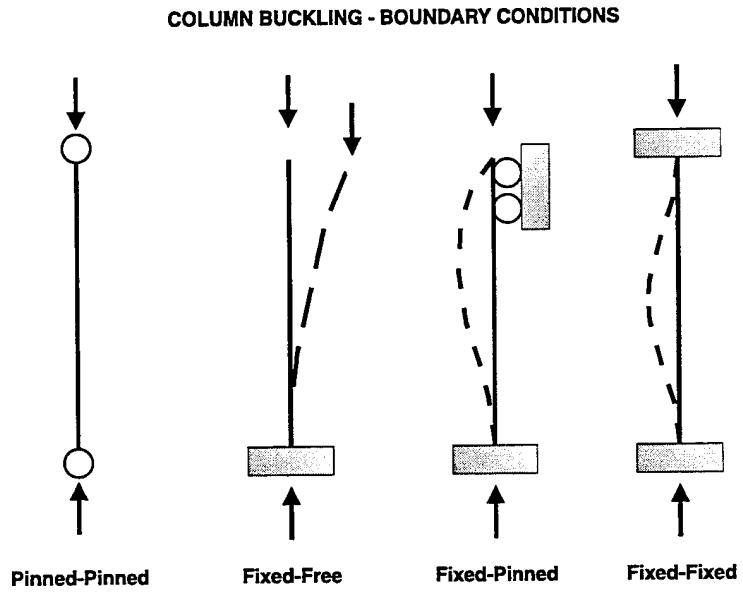
$$P_{cr} = \frac{\pi^2 E I}{L^2} \tag{4-50}$$

where I is the bending moment of inertia, A the cross-sectional area, and L the length of the column. The bucking load is then defined as:

$$\lambda = \frac{P}{P_{cr}} \tag{4-51}$$

4.5.4.1. Sensitivity Analysis

The design constraints on the buckling load factor are then written as:



$$g_{lower} = \frac{\lambda_{req}}{\lambda} - 1.0 \leq 0.0 \tag{4-52a}$$

$$g_{upper} = 1.0 - \frac{\lambda_{req}}{\lambda} \leq 0.0 \tag{4-52b}$$

where λ_{req} is the desired ratio of the load to the critical load load. Equation (4-50) is valid for a single boundary condition: pinned-pinned. There are four boundary conditions which must be considered as illustrated in the following Figure.

A summary of the critical load equations for each of these conditions is:

$$P_{cr} = \frac{\pi^2 E I}{L^2} \text{ (Pinned-Pinned)} \tag{4-53a}$$

$$P_{cr} = \frac{\pi^2 E I}{4 L^2} \text{ (Fixed-Free)} \tag{4-53b}$$

$$P_{cr} = \frac{2.05\, \pi^2 E I}{L^2} \text{ (Fixed-Pinned)} \tag{4-53c}$$

$$P_{cr} = \frac{4 \pi^2 E I}{L^2} \text{ (Fixed-Fixed)} \tag{4-53d}$$

4.5.5. Model Characteristic Constraints

The structural design task requires that limits be placed on the values over which the characteristics of the finite element model can range. ASTROS makes provision for two such characteristics, weight and thicknesses. In this discussion, these limits are generically identified as weight and thickness constraints, but the term also applies to limits on the total mass, cross-sectional areas and concentrated mass variables listed in Table 6.

Without these limits on thickness, the optimization algorithm could take the thickness to unrealistically small (or even negative) values. Unrealistically large values (e.g., thicknesses greater than the available wing depth) could also occur. Thickness constraints are specified in one of four ways, as specified in the following paragraphs.

4.5.5.1. Weight

The weight or mass characteristic is typically the objective function in ASTROS. However, it may be used as a constraint or as a constituent of a synthetic function, as well. Limits on this value allow the designer to perform optimal trade studies in which the weight plays a secondary role, rather than the primary role of the objective. For weight or mass to be used as a constraint, it must be used in a synthetic function. Weight is always the ASTROS default objective function.

4.5.5.2. Side Constraints

For the unique and the regional linking options (options 1 and 2 of the Global Variables discussion of Subsection 4.4.2), the global variables are explicitly constrained. Physical limits, manufacturing considerations or limits specified by factors not considered in ASTROS (e.g. fatigue or buckling) can all contribute to defining these constraints.

4.5.5.3. Thickness Constraints

When the shape function design variable linking option is used, side constraints on the global design variables cannot be used. Move limits on the physical design variable (local variable) are instead applied through the definition of thickness constraints. The value of the thickness constraint is determined by the user specified move limit or by the true physical upper or lower bound gauge constraints. Subsection 3.2 of the Applications Manual discusses the use of the DCONTHK data entry to explicitly select elements whose thickness constraints will always be retained in the design task. Note that the ASTROS procedure automatically generates thickness constraints for all local design variables linked to shape functions.

4.5.5.4. Move Limits

The user should be aware of a third type of thickness constraint that is internal to the ASTROS procedure. Approximation concepts (see Subsection 13.2) are based on the assumption that many response quantities are a linear function of the design variables, or their inverse. In order to maintain the validity of this approximation, limits are placed on how much a local design variable can change during a design cycle. A MAPOL parameter controls these limits, with a halving or doubling of a thickness typically permitted. These limits will be most pronounced when a user's initial design is far from the optimum. Progress toward the optimum may appear slow in these cases because the move limits are artificially restricting the design.

4.5.5.5. Laminated Composite Constraints

Manufacturing constraints for composites have also been incorporated in ASTROS. A set of design constraints handles the common composite manufacturing requirements that are applied during the design of structures using composite laminates. These constraints are on the ply minimum gauge, laminate minimum gauge and the percentage of the laminate that is made up of a ply. Together with the use of design variable linking and side constraints on both the local and global design variables, these constraints allow the user to ensure that the optimization process will generate a thickness distribution that is at least reasonably close to a manufacturable laminate. Of course, the restriction of a continuous design variable approximation to the discrete layer thicknesses is still present.

The laminate composition constraints allow the specification of an upper or lower bound on the percentage of a *laminate* thickness that is composed of a particular *ply* thickness.

$$\frac{\%_{req}}{100} - \frac{t_{ply}}{t_{lam}} \leq 0 \quad \text{(lower bound)} \tag{4-54a}$$

$$\frac{t_{ply}}{t_{lam}} - \frac{\%_{req}}{100} \leq 0 \quad \text{(upper bound)} \tag{4-54b}$$

Similarly, the lower bound constraints on the ply and laminate minimum guages are given by:

$$1.0 - \frac{t_{lam}}{t_{\min}} \leq 0 \text{ (for laminate)} \qquad 1.0 - \frac{t_{ply}}{t_{\min}} \leq 0 \text{ (for ply)} \tag{4-55}$$

All these constraints share common definitions of the *ply* and the *laminate*, which are worthy of some discussion. A *ply* consists of one or more layers whose ***summed*** thicknesses will constitute the ply thickness. Similarly, a *laminate* is one or more layers whose ***summed*** thicknesses will constitute the total laminate thickness. The principal difference between a *ply* and a *laminate* is that the defaults are set up to allow simple input when a ply is a single layer and when a laminate consists of all layers.

The generality of the *ply* and the *laminate* definitions provides for the generality of the typical composite composition constraints. For example, there are usually both lower and upper bound fractions of the *laminate* thickness for each distinct fiber orientation. These orientations may be (and usually are) separated into noncontiguous layers on the PCOMP entry. Therefore, the *ply* definition needs to be all layers with the same orientation and not just a single layer. For sandwich composites with composite face sheets, each face sheet independently has the manufacturing composition constraint imposed and the core thickness and the other face sheet are not desired in the definition of the laminate thickness. All these scenarios are accommodated in the ASTROS constraint definitions.

4.5.5.6. Thickness and Laminate Sensitivities

Thickness and laminate constraints are the only constraints in ASTROS that are direct functions of the design variables. As indicated in Subsection 4.3, the local variable is an algebraic sum of the global variables, so the j^{th} thickness constraint can be written in terms of the global design variables as:

$$g_k = 1.0 - \sum_{i=1}^{ndv} P_{ij} \frac{v_i}{t_{\min}} \quad \text{and} \quad g_k = 1.0 - \sum_{i=1}^{ndv} P_{ij} \frac{v_i}{t_{\min}} \tag{4-56}$$

the sensitivity of this constraint to the j^{th} design variable is then simply:

$$\frac{\partial g_k}{\partial v_j} = -\frac{P_{ij}}{t_{\min}} \quad \text{and} \quad \frac{\partial g_k}{\partial v_j} = +\frac{P_{ij}}{t_{\max}} \tag{4-57}$$

Similarly for laminate constraints, the design variable linking relationship is used to develop the sensitivities of these constraints. For laminate minimum guage and maximum guage constraints the expression is identical to a maximum or minimum thickness — only the number of plies in the summation causes a distinction. For laminate composition constraints, the expression can be derived by differentiating 4-54a:

$$\frac{\%_{req}}{100} - \frac{t_{ply}}{t_{lam}} \le 0 \tag{4-58a}$$

$$\frac{\partial g}{\partial v_j} = \frac{-t_{lam} \dfrac{\partial t_{ply}}{\partial v_j} + t_{ply} \dfrac{\partial t_{lam}}{\partial v_j}}{t_{lam}^2} \tag{4-58b}$$

where:

$$\frac{\partial t_{ply}}{\partial v_j} = \sum_i P_{ij} \quad \text{and } i \in \{\text{set of layers in the ply}\}$$

$$\frac{\partial t_{lam}}{\partial v_j} = \sum_i P_{ij} \quad \text{and } i \in \{\text{set of layers in the laminate}\}$$

A similar expression results for the upper bound.

4.6. SYNTHETIC FUNCTIONS IN OPTIMIZATION

ASTROS supports a general capability which allows the user to define an algebraic combination of solution results to form a ***synthetic function***. Such functions may play the role of the objective function, or they may be used as design constraints during optimization. Functions may be defined using standard arithmetic operators, trigonometric functions and logarithmic functions. They may also include intrinsic functions which are used to retrieve responses and model characteristics from the ASTROS design and analysis model.

The response intrinsics are functions that, when called with a set of parameters, or arguments, return the value of the indicated analytical response or model characteristic. For example:

```
A = STRESS(100,SIGX,,1001)
B = THICK(2001,7)
```

returns the σ_x component of stress for element 100, case 1001 into A and the thickness of the seventh layer of element 2001 into B. In general, the synthetic constraint or objective function may be defined as:

$$a_s = h(r_1, r_2, \ldots, r_n) \tag{4-59}$$

where:

a_s is the synthetic function value

h is some set of algebraic operators

r_i are responses of the finite element analysis and/or model characteristics

In ASTROS, a_s is always a scalar function, and each of the operators, h, eventually operates on a scalar response, r_i.

When a_s is used as the objective function, there are no special requirements other than that a_s be a once differentiable function of the design variables. When a_s is used as a constraint function, the additional requirement that $a_s \leq 0$ indicate feasibility — or satisfaction of the constraint — is imposed.

Given that a_s is used in either role in the optimization process, the derivative of a_s with respect to the design variables is required. To obtain this derivative, ASTROS uses the chain rule:

$$\frac{\partial a_s}{\partial v_j} = \bar{h}\left(r_1, \frac{\partial r_1}{\partial v_j}, r_2, \frac{\partial r_2}{\partial v_j}, \ldots, r_n, \frac{\partial r_n}{\partial v_j}\right) \tag{4-60}$$

The function $\bar{h}$ is determined by the ***symbolic differentiation*** of h. The user does not need to describe $\bar{h}$ to ASTROS.

The intrinsic functions available in ASTROS are quantities such as stress, strain and displacement — the same quantities that are used in the assembly of other constraint and objective functions. The derivatives of these responses, which are used to compute the sensitivities of the synthetic functions, are presented in subsequent sections of this manual with the analytical disciplines that generate the associated response.

4.7. SENSITIVITY ANALYSIS

Mathematical programming approaches to the solution of Equations 4-1 through 4-4 typically require the gradients of the objective and the constraints with respect to the design variables. That is:

$$\begin{aligned} &\frac{\partial F}{\partial v_i} \quad i = 1, \ldots ndv \\ &\frac{\partial g_j}{\partial v_i} \quad j = 1, \ldots ncon;\ i = 1, \ldots ndv \end{aligned} \tag{4-61}$$

Previous procedures similar to ASTROS have used one of two approaches to supply these gradients: (1) finite difference analyses and (2) analytical analyses. The first approach calculates the gradients by making a perturbation in the design variable, reanalyzing the problem and computing the gradients based on:

$$\frac{\partial g_j}{\partial v_i} = \frac{g_j\left(v_i + \Delta v_i\right) - g\left(v_i\right)}{\Delta v_i} \tag{4-62}$$

The TSO procedure of Reference 4 uses this technique. Finite difference calculations become burdensome when there are large numbers of design variables and constraints, so a significant effort was expended in the ASTROS procedure to provide analytical gradient information.

The constraint and objective function gradients are computed by differentiation of the equations of motion for the particular analysis discipline and constraint (or objective). These equations are described with their associated constraint functions in Sections 6, 7, 9, and 10; but, they all eventually rely on the existence of the first order gradients of the elemental matrices with respect to the physical (or local) design variable.

Local design variables in ASTROS are quantities like element thickness (for 2-D elements), and cross-sectional areas of 1-D elements. An analytic gradient of the elemental matrices is possible for some combinations of elements, constraints and local design variables. Under these circumstances, ASTROS uses this purely analytical approach, since it is computationally efficient. When the combination of design variable and elemental properties result in a nonlinear relationship, ASTROS uses finite difference methods to compute the elemental matrix derivatives. These derivatives are then used in the analytical expression for the constraint or objective function gradient. This approach is called a Semi-Analytical method for constraint gradient computation.

4.7.1. Linear Design Variables

The term ***linear design variable*** is used to denote a combination of local design variable and elemental properties that yield a factorable, linear relationship among the design variable and the elemental matrices. For example, a rod element with no torsional stiffness, no nonstructural mass with a cross-sectional area design variable, has a linear relationship between the area and the stiffness and between the area and mass. When stress or strain constraints are applied in ASTROS, a further relationship is needed: that between the stress or strain and the nodal displacements (refer to Section 5.3 for details). This matrix is referred to as the $\boldsymbol{S}$ matrix. Again, ASTROS attempts to factorize the resultant matrices whenever possible.

The combination of these three elemental matrix quantities, mass, stiffness and stress/displacement determine whether a design variable is linear for that element. For linear design variables, the trio of design invariant elemental matrix factors are computed once. Matrix assembly and sensitivity analysis is then performed using these factors.

4.7.2. Nonlinear Design Variables

The term ***nonlinear design variable*** is used to denote a combination of local design variable and elemental properties that does not yield a factorable relationship among the design variable and the elemental matrices. For example, a plate bending element has no factorable relationship between thickness and stiffness.

For nonlinear design variables, the trio of elemental matrices must be recomputed at each new design point. Also, since no analytical gradient for these matrices is readily available, ASTROS uses finite difference methods (at the elemental level) to compute the approximate first derivative of these elemental matrices with respect to each of the local design variables connected to the finite element. This is subtly different than in the linear case, where the factored matrices can play a role in matrix assembly and sensitivity analysis. For the nonlinear case, separate matrices are needed for assembly of the global matrix and for sensitivity analysis AND both sets of matrices must be recomputed at each new design point.

For example, considering a single local design variable, t; the elemental stiffness, mass and $\boldsymbol{S}$ matrices are computed for the current value of the local design parameter, t_n. Then, the derivative of the elemental matrices can be formed by finite difference:

$$\left.\frac{\partial \boldsymbol{K}_{ee}}{\partial t_i}\right|_n = \frac{\left.\boldsymbol{K}_{ee}\right|_n (t_i + \Delta t_i)_n - \left.\boldsymbol{K}_{ee}\right|_n}{\Delta t_i} \tag{4-63a}$$

$$\left.\frac{\partial \boldsymbol{M}_{ee}}{\partial t_i}\right|_n = \frac{\left.\boldsymbol{M}_{ee}\right|_n (t_i + \Delta t_i)_n - \left.\boldsymbol{M}_{ee}\right|_n}{\Delta t_i} \tag{4-63b}$$

$$\left.\frac{\partial \boldsymbol{S}}{\partial t_i}\right|_n = \frac{\left.\boldsymbol{S}\right|_n (t_i + \Delta t_i)_n - \left.\boldsymbol{S}\right|_n}{\Delta t_i} \tag{4-63c}$$

In ASTROS, the updated matrices and finite difference sensitivities are only computed for non-linear designed elements. These data are combined with design invariant and linear designed elements to assemble both the updated global matrices and the updated first order gradient matrices.

$$\boldsymbol{K}_{gg} = \mathbf{A}\, \boldsymbol{k}^{ee}_{fixed} + \mathbf{A} \sum_j \sum_i p_{ij} \boldsymbol{k}^{i}_{fact} v_j + \mathbf{A}\, \boldsymbol{k}^{ee}_{nl} \tag{4-64a}$$

$$\boldsymbol{M}_{gg} = \mathbf{A}\, \boldsymbol{m}^{ee}_{fixed} + \mathbf{A} \sum_j \sum_i p_{ij} \boldsymbol{m}^{i}_{fact} v_j + \mathbf{A}\, \boldsymbol{m}^{ee}_{nl} \tag{4-64b}$$

$$\boldsymbol{S} = \mathbf{A}\, \boldsymbol{s}_{fixed} + \mathbf{A}\, \boldsymbol{s}_{nl} \tag{4-64c}$$

$$\frac{\partial \boldsymbol{K}_{gg}}{\partial v_j} = \mathbf{A} \sum_i p_{ij} \boldsymbol{k}^{i}_{fact} + \mathbf{A} \sum_i p_{ij} \frac{\partial \boldsymbol{k}_{ee}}{\partial t_i} \tag{4-64d}$$

$$\frac{\partial \boldsymbol{M}_{gg}}{\partial v_j} = \mathbf{A} \sum_i p_{ij} \boldsymbol{m}^{i}_{fact} + \mathbf{A} \sum_i p_{ij} \frac{\partial \boldsymbol{m}_{ee}}{\partial t_i} \tag{4-64e}$$

$$\frac{\partial \boldsymbol{S}}{\partial v_j} = \mathbf{A} \sum_i p_{ij} \frac{\partial \boldsymbol{s}_{ee}}{\partial t_i} \tag{4-64f}$$

where the global matrix is assembled from the combination of 1) design invariant elemental matrices, 2) elemental matrices that represent factorized "derivatives" and 3) recomputed elemental matrices. The global matrix sensitivities are assembled from 1) the factorized linear derivatives and 2) the finite difference derivatives for nonlinear design variables. Note that Equation 4-6 has been used to account for linking during the assembly process. This is actually what occurs in ASTROS, since the finite difference takes place on the local design variable.

To simplify the remainder of this manual, the effects of nonlinear design variables will be embedded in the first order sensitivity matrices and the global matrices. Unless specifically required in context, no further distinctions will be made in first order gradient computations for linear and nonlinear design variables.

THIS PAGE INTENTIONALLY LEFT BLANK

5. FINITE ELEMENT MODELING

This section provides a description of the finite elements available in ASTROS and the algorithms used to assemble the individual elements into global mass and stiffness matrices. Emphasis is placed on the design aspects of this modeling. Section 5 of Reference 1 and Section 8 of Reference 2 contain thorough discussions of the elements used in ASTROS. An exception to this is a complete description of the **QUAD4** element in Appendix A. This element was developed specifically for the ASTROS program and therefore requires detailed documentation.

5.1. FINITE ELEMENTS

A limited set of elements have been implemented in ASTROS. The selection of the elements was based primarily on past experience in the modeling of aerospace structures. Another consideration was the selection of elements that lend themselves to an automated design task. The discussion which follows divides the elements into seven categories:

- Concentrated mass elements
- Scalar elements
- One-dimensional elements
- Two-dimensional elements
- Three-dimensional elements
- The General element
- Rigid elements

The discussion in this subsection is primarily devoted to the formation of the element stiffness and mass matrices and the thermal load sensitivity vectors (referred to below as the thermal vectors). Subsection 5.2 contains a discussion of the calculation of stress and strain constraints for the ASTROS' elements while Subsection 7.2.1 of the User's Manual discusses the output that is available for each of the elements.

5.1.1. Concentrated Mass Elements

These elements allow for the definition of mass properties without any associated stiffness. They are useful for modeling the mass properties of a structure, which are typically defined by a separate group from that used in the structural stiffness modeling. In the design context, these elements can be used by the design to size tuning masses when frequency constraints are to be satisfied or as a mass balance variable in a flutter design task.

ASTROS has provided two separate forms for specifying concentrated masses that have been adapted from the **CONM1** and **CONM2** elements of NASTRAN. In the **CONM1** form, the user inputs the top half of a symmetric mass matrix at a geometric grid point. The bulk data entry for this element in the User's Manual completely defines its form. The element cannot be designed.

In the **CONM2** form, the user inputs mass data about a center of gravity point that may be offset from a geometric grid point. The mass matrix at the grid point is then calculated using:

$$\boldsymbol{m} = \overline{m}\begin{bmatrix} 1 & 0 & 0 & 0 & z & -y \\ & 1 & 0 & -z & 0 & x \\ & & 1 & y & -x & 0 \\ & & & y^2+z^2 & -xy & -xz \\ & SYM & & & x^2+z^2 & -yz \\ & & & & & x^2+y^2 \end{bmatrix} + \begin{bmatrix} \mathbf{0} & & \mathbf{0} & \\ & I_{11} & -I_{21} & -I_{31} \\ \mathbf{0} & -I_{21} & I_{22} & -I_{32} \\ & -I_{31} & -I_{32} & I_{33} \end{bmatrix} \tag{5-1}$$

where x, y, and z are the offset distances from the mass to the associated grid point in a specified coordinate system, $\overline{m}$ is the mass value and the $\boldsymbol{I}_{ij}$ terms are inertias about the mass center of gravity. The mass matrix of Equation 5-1 is in the input coordinate system. It may be necessary to make a coordinate transformation to the global coordinate system. Equation 5-1 is written in the form it is to stress the point that the design variable for ASTROS for this element is the $\overline{m}$ value and that the input inertia terms remain unchanged when design is being performed. These design features give an element mass matrix that is linear in the design variable if $\boldsymbol{I}_{ij} = 0$ or is separable into a constant term and linear term.

There are no thermal effects or output recovery for these elements.

5.1.2. Scalar Elements

The scalar spring element and the scalar mass element, based on the **CELAS** and **CMASS** elements of NASTRAN, are available in ASTROS. The element matrices for these items are

$$\boldsymbol{k} = \overline{k}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{5-2}$$

and

$$\boldsymbol{m} = \overline{m}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{5-3}$$

where $\overline{m}$ and $\overline{k}$ are user input values and the matrices are in relation to the two degrees of freedom specified by the user. Both $\overline{m}$ and $\overline{k}$ can be design variables in ASTROS, although the physical meaning of the scalar mass variable is not clear and its use appears limited. The spring variable can be used to represent, for example, an actuator stiffness which could be included in the design process. Note that the sensitivity of the weight to changes in the scalar spring design variable is zero. This presents no

particular difficulty, but it may result in a poorly posed problem if the user naively assumes that infinite stiffness is achievable in a real-world situation for no penalty.

There are no thermal effects for these elements, nor are there any stress constraints in the design task. The user can, however, impose displacement constraints which may emulate a stress constraint for the scalar spring element.

5.1.3. One-Dimensional Elements

Two one-dimensional elements, the rod and the bar, have been implemented in ASTROS. These are described in the following sections.

5.1.3.1. The Rod Element

The rod element of Figure 11 has both extensional and torsional stiffness with an assumed linear displacement field. This field gives rise to constant element stresses. The implementation of this element has been based on that used for the **CROD** (or **CONROD**) element in NASTRAN. As the figure indicates, the rod has two degrees of freedom at each node in its element coordinate system. After transformation to the global coordinate system, this results in either a 6 x 6 or 12 x 12 element stiffness matrix, depending on whether the user has specified only extensional or both torsional and extensional stiffness values. The mass matrix is always 6 x 6 since only the translational degrees of freedom have inertia properties associated with them. The user is given the option as to whether a lumped or a consistent formulation of the mass matrix is to be used. The thermal vector is 6 x 1.

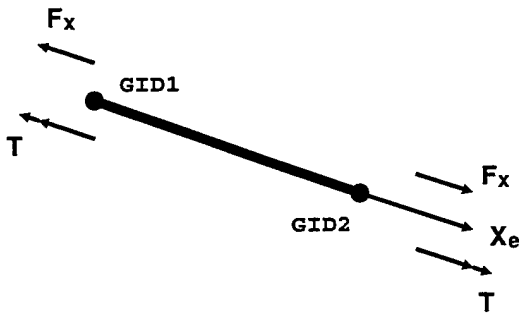


Figure 11. The ROD Element

The design variable for the rod element is its area. Two modifications to the element matrix calculations are made if the element is designed. The first is that user input values of the torsional stiffness are used to generate a separate, nondesigned pseudo-element that has only the torsional stiffness. This is done because there is no general relationship between the rod area and the torsional stiffness. The second modification is that user input values for the nonstructural mass are used to generate a mass matrix for the pseudo-element which has only the design invariant nonstructural mass. The nonstructural masses for all elements are treated in a similar manner. This ensures that the element mass matrix due to its volume is a linear function of the design variable.

Stress constraint calculations for designed rods include shear stresses since the torsional stiffness has been retained. The torsional stiffness, however, is not a function of the design variable (area) and so the constraint may not be controlled by the design variables. If the user specifies stress constraints on an element that is not designed and that has torsional stiffness, shear stresses are also included in the constraint calculation.

5.1.3.2. The BAR Element

The bar element of Figure 12 includes extension, torsion and bending in two perpendicular planes with associated transverse shear properties. The bar element has the following modeling features and limitations, as given in Subsection 9.5 of Reference 14:

1. The neutral axis may be offset from the grid points.
2. The neutral axis and shear center coincide.
3. Pinned connections may be defined.
4. The area properties are constant.
5. The principal axes of inertia need not coincide with the local axes.
6. Stress can be recovered at four points on the cross section on each end.

Pinned connections allow the specification of degrees of freedom that cannot transfer force, thereby creating a hinge. As Figure 12a indicates, six degrees of freedom are present at each node, resulting in a 12 x 12 element stiffness matrix. Rows and columns associated with pinned degrees of freedom are zeroed out. The element mass matrix is also 12 x 12 and has off-diagonal terms if a consistent mass formulation is used or if the beam is offset from the grid points. The thermal vector is 12 x 1 and thermal gradients are neglected (i.e., the **TEMPBR** data entry of NASTRAN is not supported).

Unlike the other ASTROS elements, the bar element may have either one design variable, its cross-sectional area, or it may have several design variables which represent parameters in a family of standard cross-sections. If the element is designed using a single variable, a number of restrictions are placed on the modeling. As in the rod element, user specifications of nonstructural mass are used to spawn a design invariant pseudo-element.

Using a Single Design Variable

When using a single design variable for the bar element, there is a fixed relationship between the cross-sectional area and moments of inertia of the element. This was introduced in Equation 2-5 and is repeated here:

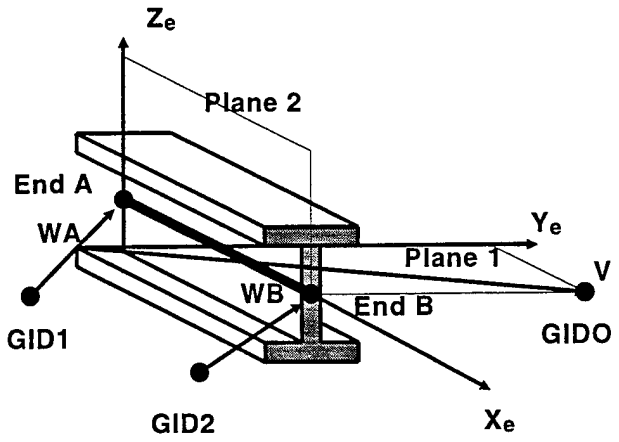
$$I_1 = r_1 A^{\alpha}$$
$$I_2 = r_2 A^{\alpha} \tag{4-5}$$

Implementation of the relationships of Equation 2-5 dictates that the total element stiffness matrix is made up of a term that is linear in the cross sectional area and a second term that is this same area raised to an exponential power:

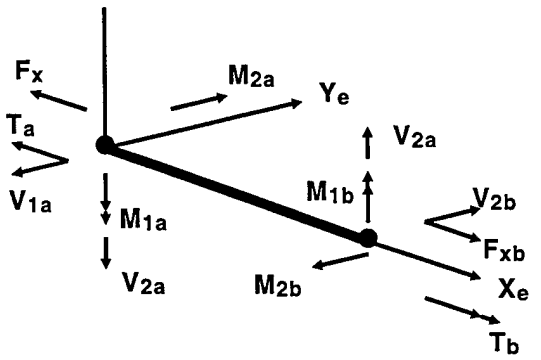
$$\boldsymbol{K}_e = A\ \boldsymbol{K}_e^E + A^{\alpha}\ \boldsymbol{K}_e^R \tag{5-4}$$

where the E and R superscripts refer to extensional and rotational stiffness terms, respectively.

When modeling the single design variable, the **PBAR** Bulk Data entry is used for defining the element property and the inertia linking terms. In this case, input shear factors and cross products of



a. Element Coordinate System



b. Element Forces and Sign Conventions

Figure 12. The BAR Element

inertia values are set to zero. Neither the pin connection feature nor the offset feature is supported for a designed beam.

Limitations imposed on designed bar elements are imposed on the stress computations for that element as well. If, however, the user specifies stress constraints on an element that is not designed, the full finite element capabilities are included in the stress computation.

Using Cross-Sectional Design Variables

By using the **PBAR1** Bulk Data entry, the user may select from a number of standard cross-sections as shown in Figure 13. These sections are defined using from one to six geometric parameters, each of which becomes a design variable. The user may then specify all appropriate side constraints on each variable, and may link variables within the element. No limitations are imposed in the properties of a designed bar when the **PBAR1** Bulk Data entry is used. The matrix sensitivities are computed using finite difference methods.

The four stresses computed at the nodes of the bar elements are axial stresses due to axial strain and bending. Stresses are recovered at the locations specified in the element coordinate system which move in relation to the design variables.

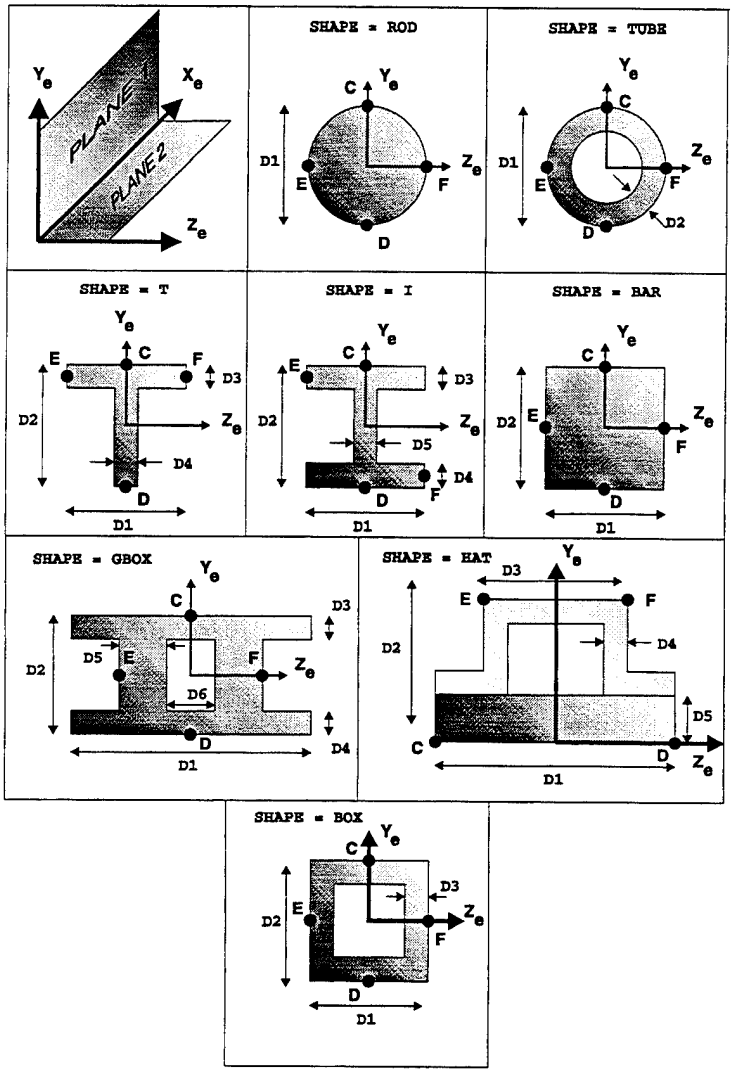


Figure 13. The PBAR1 Standard Sections

5.1.4. Two-Dimensional Elements

Five two-dimensional elements are available in ASTROS: a quadrilateral shear element (SHEAR), a triangular membrane element (TRMEM), a quadrilateral membrane element (QDMEM1), a quadrilateral bending element (QUAD4), and a triangular bending element (TRIA3). The quadrilateral and triangular bending elements are similar to the UAI/NASTRAN QUAD4 and TRIA3 elements. These are the prefered modeling elements since their shear and membrane capabilities are equivalent to the SHEAR and QDMEM1/TRMEM elements.

5.1.4.1. The Quadrilateral Shear Element

The shear element shown in Figure 14 is a two-dimensional quadrilateral element that resists only in-plane shear forces. The element is defined relative to a mean plane parallel to the plane of the diagonals and located midway between them. Garvey's assumption that the shear flow distribution is constrained to satisfy equilibrium conditions, with no requirement on strain compatibility, is used (See Subsection 5.3 of Reference 1). This assumption is exact for rectangular elements and becomes more approximate as the distortion from this rectangular shape increases.

Element stiffness and mass matrices of dimension 12 x 12 are generated for the translational degrees of freedom. Only isotropic material properties are implemented for this element and only a lumped formulation of the mass matrix can be computed. No temperature effects are included in this element.

The design variable for the shear panel is the element thickness. If the element is designed, user specified values of the nonstructural mass for the element are used to generate a design invariant pseudo-element's mass matrix. Stress constraints for the shear element are calculated based on the average of the shear stresses at the four nodes.

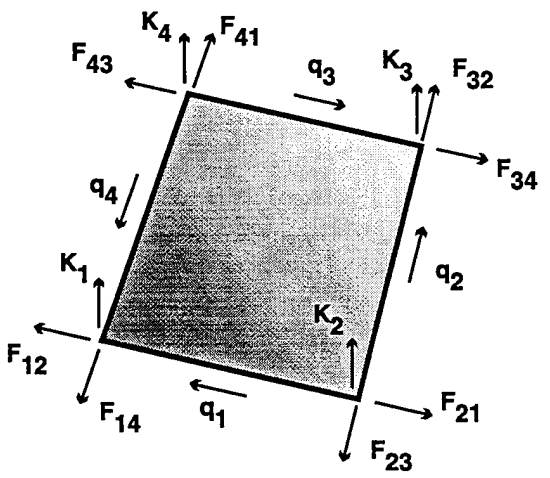


Figure 14. The Quadrilateral Shear Element

5.1.4.2. The Triangular Membrane Element

The membrane element shown in Figure 15 is a two-dimensional triangular element that resists only in-plane forces and is equivalent to the TRMEM element in NASTRAN. The displacement field is assumed to vary linearly in the coordinates of the element, giving rise to a constant strain state within the element. Both isotropic and anisotropic materials can be analyzed, with the θ angle of Figure 15 used to define the property axis for an anisotropic material.

Element stiffness and matrices are 9 x 9 for this triangular element. Only a lumped mass formulations of the mass matrix is available. The thermal vector is of dimension 9 x 1, with the thermal loading taken to be the average of the temperatures at the three element nodes.

The design variable for the triangular element is the element thickness. Separate design variables are available for each ply direction if a composite material is being designed. This is consistent with the FASTOP formulation of Reference 5, which treats all the plies within a laminate that are aligned in the same direction as a "layer." In reality, of course, ply layup ordering is of critical importance and must be considered in the detailed design of a composite part. If only membrane forces are being considered, ply order effects do not matter and the lumping of plies is permissible for analysis purposes. Ply orientation angles are not available as a design variable. However, there is no limit on the number of ply directions that a user can specify and it is conjectured that if a user selects a large number of directions (say six), a winnowing process will take place and desirable orientation directions will present themselves. If the element is designed, user specified values of the nonstructural mass are used in the usual manner to generate a design invariant portion of the mass matrix.

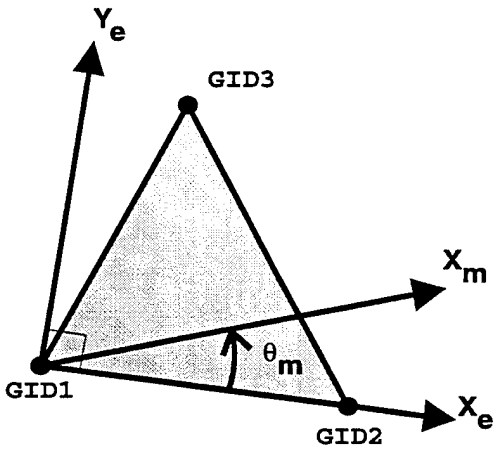


Figure 15. The Triangular Membrane Element

5.1.4.3. The Isoparametric Quadrilateral Membrane Element

The membrane element shown in Figure 16 is a two-dimensional quadrilateral element that resists only in-plane forces and is equivalent to the QDMEM1 element in NASTRAN. The element has the following properties, as discussed in Subsection 5.8.5 of Reference 1:

1. The stresses and strains vary within the element in an essentially linear manner.
2. The element may have a warped shape; i.e., the four nodes need not be co-planar.
3. Gaussian quadrature, with a 4 x 4 grid, is used to evaluate the stiffness matrix.
4. The temperature is assumed to be constant over the element, and is the average of the nodal temperatures.

The "isoparametric" designation refers to the fact that the equation which relates the displacements at any point in the element to the displacements at the nodes in terms of parametric coordinates (ξ , η) is identical in form to the equation which relates the internal coordinates to the coordinates of the grid points. Both isotropic and anisotropic materials can be accommodated by the element, with a material axis defining the orientation of the anisotropic properties.

The element grid points are mapped to a mean plane located midway between the diagonals of the element, resulting in a planar quadrilateral. The 12 x 12 stiffness matrix is then derived for this quadrilateral and then transformed to the physical grid points. The 12 x 12 mass matrix is calculated using a lumped formulation. The thermal vector is of dimension 12 x 1.

The design variable for the quadrilateral element is the element thickness. Separate design variables are available for each ply direction if a composite material is being designed. The comments on

composite design just discussed for the triangular element apply to this element as well. If the element is designed, user specified values of the nonstructural mass generate the appropriate design invariant portion of the mass matrix.

5.1.4.4. The Shell Elements

The QUAD4 and TRIA3 elements in ASTROS are provided to allow for the inclusion of bending effects in quadrilateral and triangular elements and to give a general treatment of composite materials. Appendix A provides a detailed theoretical treatment of this development, with an overview provided here. The geometry of these elements is shown in Figures 17 and 18.

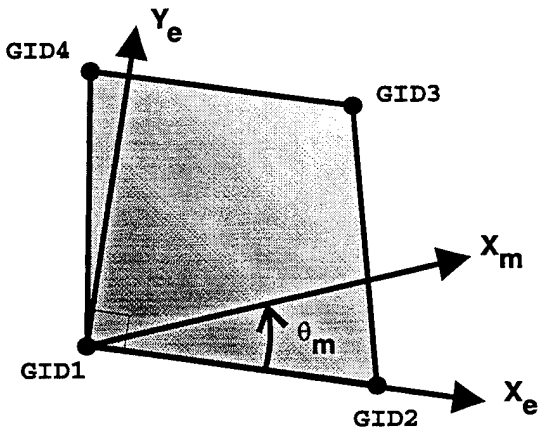


Figure 16. The Quadrilateral Membrane Element

The formulation for these isoparametric elements incorporates a bilinear variation of geometry and deformation within the element. Both the QUAD4 and TRIA3 elements have 5 degrees of freedom (DOF) per node, i.e., the stiffness for rotation about the normal to the mid-surface at each node is not defined. Furthermore, it is assumed that plane sections remain plane and that the variation of strains through the thickness is linear. Direct strain through the thickness is neglected (assumed to be zero).

Both elements may be used to model either membrane or bending behavior, or both. Transverse shear flexibility may be requested as well as the coupling of membrane and bending behaviors using nodal offsets or linear variation of material properties through the thickness. In addition, the QUAD4

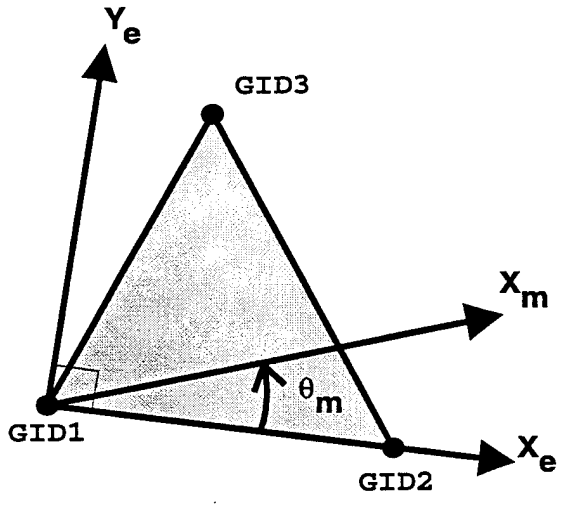


Figure 17. The TRIA3 Element Geometry

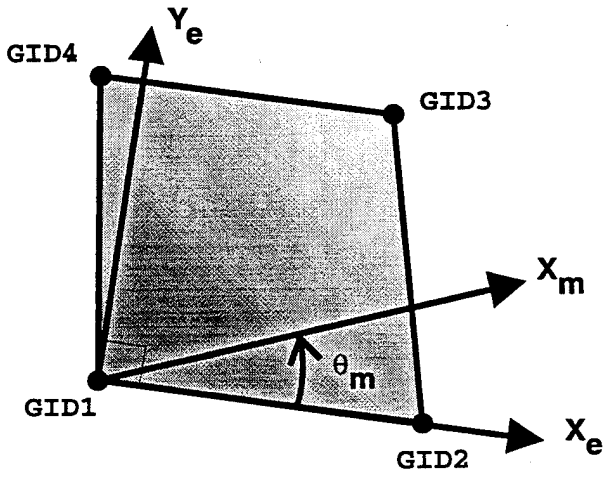


Figure 18. The QUAD4 Element Geometry

and TRIA3 elements are capable of representing laminated composite materials, with an option to compute interlaminar shear stresses and layer failure indices.

The transverse shear stiffness is numerically conditioned to enhance the accuracy of the element for a wide range of modeling practices including very thick or thin elements, high aspect ratio elements and skewed elements. Numerical conditioning of the out-of-plane shear strains is discussed in Appendix A.

The elements provide lumped or, optionally, consistent mass matrices. The equivalent pressure and/or thermal loads are also calculated. Thermal effects are accounted for in the element stress and force recovery.

Design sensitivity matrices, constraints and gradients of constraints are computed for use in the structural optimization procedure. The element thickness or, for composites, individual layer thickness are the design variables. When elemental properties are nonlinear in the design variables (such as when bending properties exist), finite difference methods are used at the elemental level to compute the sensitivity of stiffness, mass and stress, or strain/displacement matrices, to the design variable. In the latter quantity, care is taken to account for any movement of the stress computation points during a change in the design variable.

5.1.5. Solid Elements

Three isoparametric hexahedron solid elements have been implemented in ASTROS. These are the 8 noded IHEX1, the 20 noded IHEX2, and the 32 noded IHEX3. These three elements are essentially identical to the COSMIC/NASTRAN elements of the same names (see Subsection 5.13 of Reference 1). Typically, the IHEXi elements would be used to model geometrically complex thick-walled components.

Solid elements cannot be designed due to their not having any dimensional parameters such as thickness or cross sectional area, which can be modified without violating inter-element compatibility. Nevertheless, these elements may still be utilized in optimization runs although they, themselves, will not be designed.

The "isoparametric" designation follows from the fact that the same interpolating functions are used for both the element geometry and the element deformation. The interpolating functions are either linear, quadratic, or cubic, and are used to represent the IHEX1, IHEX2, and IHEX3 elements, respectively. These functions are chosen so as to ensure interelement compatibility and to satisfy the constant-strain convergence criteria.

The stiffness, mass and load equations for the IHEXi elements are derived using the principle of virtual work. The equations are then evaluated by application of Gaussian Quadrature. The number of integration points used to evaluate the stiffness, mass and load matrices defaults to 2 x 2 x 2 for the linear element, 3 x 3 x 3 for the quadratic element, and 4 x 4 x 4 for the cubic element. Optionally, other integration mesh sizes may be specified. All computations are performed in the basic coordinate system, and the resulting matrices are then transformed into the global coordinate system in preparation for the element matrix assembly.

Element stresses, strains, and strain energies are calculated based on the displacements determined in the global analysis of the structure. The stresses and strains are calculated in the basic coordinate system at the eight corner points and at the center of the element. Stresses and strains are calculated also at the center of each edge in the case of IHEX2 and IHEX3 elements. In addition, the principal stresses and strains, principal direction cosines, and mean and octahedral stresses and strains are computed at each of the above points.

5.1.6. The General Element

The general element is a stiffness element connected to any number of degrees of freedom. It is most often used to model nonstructural components such as servomechanisms for which laboratory testing has provided stiffness or flexibility data. Two options are provided to define the characteristics of such an element.

The first option requires specifying the deflection influence coefficients for the element when it is supported in a non-redundant manner. The associated matrix of the restrained rigid body motions may be specified or it may be generated by ASTROS. The second option requires the stiffness matrix of the element to be defined directly. This stiffness matrix may represent the unsupported body, containing all the rigid body modes, or it may represent a portion of the degrees of freedom from which some or all of the rigid body motions are deleted. In the latter case, an option is available to reintroduce the restrained rigid body terms, providing that the original support conditions were not redundant. An important advantage of this option is that, if the original support conditions restrain all rigid body motions, the reduced stiffness matrix need not be specified to high precision in order to preserve the rigid body properties of the element.

The force-displacement relationship for the general element, when written in the flexibility form, is:

$$\begin{Bmatrix} \boldsymbol{u}_i \\ \boldsymbol{f}_d \end{Bmatrix} = \begin{bmatrix} \boldsymbol{Z} & \boldsymbol{S} \\ -\boldsymbol{S}^T & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{f}_i \\ \boldsymbol{u}_d \end{Bmatrix} \tag{5-5}$$

where $\boldsymbol{Z}$ is the matrix of deflection influence coefficients for coordinates $\boldsymbol{u}_i$ when coordinates $\boldsymbol{u}_d$ are rigidly restrained. $\boldsymbol{S}$ is a rigid body matrix whose terms are the displacements $\boldsymbol{u}_i$ due to unit motions of the coordinates $\boldsymbol{u}_d$, when all $\boldsymbol{f}_i = \boldsymbol{0}$. $\boldsymbol{f}_i$ are the forces applied to the element at the $\boldsymbol{u}_i$ coordinates. $\boldsymbol{f}_d$ are the forces applied to the element at the $\boldsymbol{u}_d$ coordinates. They are assumed to be statically related to the $\boldsymbol{f}_i$ forces, so that they constitute a non-redundant set of reactions for the element.

In terms of stiffness coefficients, this relationship is:

$$\begin{Bmatrix} \boldsymbol{f}_i \\ \boldsymbol{f}_d \end{Bmatrix} = \begin{bmatrix} \boldsymbol{k} & -\boldsymbol{kS} \\ -\boldsymbol{S}^T\boldsymbol{k} & \boldsymbol{S}^T\boldsymbol{kS} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_d \end{Bmatrix} \tag{5-6}$$

where all symbols have the same meaning as above and $\boldsymbol{k} = \boldsymbol{Z}^{-1}$, if $\boldsymbol{k}$ is nonsingular. It is permissible for $\boldsymbol{k}$ to be singular.

Table 7. Summary of ASTROS Rigid Elements

ELEMENT	DESCRIPTION
RBAR RBE1 RBE2	Define rigid connections between GRID points using different specifications of the dependent DOF.
RBE3	Used to average the motions at a GRID point or to distribute loads among GRID points.
RROD	Defines a rigid extensional connection between two GRID points.

No internal forces or other solution results are produced for the general element, and, while it can participate in optimization models, it cannot be designed.

5.1.7. Rigid Elements

ASTROS provides you with five special *rigid elements* that may be used to conveniently model rigid body structural connections and to define motions of points as an *average* of the motion of other points. Table 7 summarizes the Bulk Data entries that define these rigid connections. The discussions of the RROD, RBAR, RBE1 and RBE2 are adopted from Section 3.5 of Ref. 1.

5.1.7.1. The RROD Element

The RROD element represents a pin-ended connection between two grid points that is rigid in extension-compression. Suppose A and B are two grid points connected by an RROD element and α_1, α_2 and α_3 are the direction cosines, with respect to the basic coordinate system, of the line joining A and B. Then since the distance between A and B remains unchanged, the following condition is satisfied for small displacements:

$$(u_A - u_B)\alpha_1 + (v_A - v_B)\alpha_2 + (w_A - w_B)\alpha_3 = 0 \tag{5-7}$$

which may be written in matrix form as:

$$[\alpha_1 \quad \alpha_2 \quad \alpha_3] \begin{Bmatrix} u_A \\ u_A \\ w_A \end{Bmatrix}_B = [\alpha_1 \quad \alpha_2 \quad \alpha_3] \begin{Bmatrix} u_B \\ u_B \\ w_B \end{Bmatrix}_B \tag{5-8}$$

Equation 5-8 may then be rewitten to account for the transformation to global coordinates as:

$$[\alpha_1 \quad \alpha_2 \quad \alpha_3] \left[T_A \right] \begin{Bmatrix} u_A \\ u_A \\ w_A \end{Bmatrix}_G = [\alpha_1 \quad \alpha_2 \quad \alpha_3] \left[T_B \right] \begin{Bmatrix} u_B \\ u_B \\ w_B \end{Bmatrix}_G \tag{5-9}$$

Equation 5-9 is the single equation of constraint that represents the a rigid pin-ended connection between the grid points A and B. Note that only the three translational components of motion at each of the two points are involved in this equation. Any one of the six translational components may be specified as the dependent degree of freedom in a RROD element. The remaining five components are treated as reference degrees of freedom.

5.1.7.2. The RBAR and RBE2 Elements

The RBAR and RBE2 elements are similar in that they both involve a single reference grid point and one or more dependent grid points. The RBAR element is the simpler. It defines a rigid connection in which up to six degrees of freedom of a single dependent grid point are coupled to six degrees of freedom of the independent grid point. The RBE2 element is more general. It defines a rigid connection in which selected degrees of freedom of the dependent grid points are coupled to all six degrees of freedom of the independent point.

Consider a dependent grid point A that is rigidly coupled to grid point B by an RBAR element. For small displacements, the motion u_A at point A is related to that of point B, u_B , by:

$$\begin{Bmatrix} u_{A_1} \\ u_{A_2} \\ u_{A_3} \\ u_{A_4} \\ u_{A_5} \\ u_{A_6} \end{Bmatrix}_G = \begin{bmatrix} T_A^T & 0 \\ 0 & T_A^T \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & (z_B - z_A) & -(y_B - y_A) \\ 0 & 1 & 0 & -(z_B - z_A) & 0 & (x_B - x_A) \\ 0 & 0 & 1 & (y_B - y_A) & -(x_B - x_A) & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T_B & 0 \\ 0 & T_B \end{bmatrix} \begin{Bmatrix} u_{B_1} \\ u_{B_2} \\ u_{B_3} \\ u_{B_4} \\ u_{B_5} \\ u_{B_6} \end{Bmatrix}_G \tag{5-10}$$

which may be written in the more compact form:

$$\boldsymbol{u}_A = \boldsymbol{G}_{AB}\, \boldsymbol{u}_B \tag{5-11}$$

where $\boldsymbol{G}_{AB}$ is a 6x6 matrix. Each row of this matrix corresponds to a dependent degree of freedom of grid point A, and each column corresponds to a reference dgree of freedom of grid point B. Each element of this matrix represents a coefficient that corresponds to the coupling of a particular dependent degree of freedom of A with a particular reference degree of freedom of B.

Equation 5-11 defines a set of six linear equations of constraint that mathematically represent the rigid coupling of dependent grid point A to reference grid point B. In the case of an RBAR element, six equations of motion are generated relating any arbitrary set of DOF from A and/or B, provided that they fully describe the rigid body motion.

In the case of the RBE2 element, Equation 5-10 is assembled for a set of dependent grid points. The six specified dependent DOF of these grid points form the rows of the relationship given in Equation 5-11 where $\boldsymbol{u}_A$ now represents degrees of freedom from multiple reference grids.

5.1.7.3. The RBE1 Element

The RBE1 element is the most general rigid element. It defines a rigid connection in which selected degrees of freedom of the dependent grid points are coupled to six selected reference degrees of freedom. The six reference degrees of freedom can be selected at one or more (up to six) reference grid points, but they must, together, be capable of fully describing rigid body motion.

Let B be one of the reference grid points in a RBE1 element and let m be the total number of dependent degrees of freedom specified on the element. Then, for small displacements, the m equations of constraint may be expressed in terms of the motion of grid point B as defined in Equation 5-11. Note, however, that in the case, as with the RBE2, the six degrees of freedom at grid point B will not typically all be the required reference degrees of freedom. Using the user-defined set of dependent degrees of freedom, Equation 5-11 is assembled for all combinations. Then, the rows and columns associated with the specified degrees of freedom are used to assemble the constraint equations.

5.1.7.4. The RBE3 Element

The RBE3 element differs from the preceding true rigid elements in that it represents a distribution element with zero or limited stiffness, rather than infinite stiffness (Reference 36). The formulation of the RBE3 is performed by assuming the motions of the dependent degrees of freedom are a weighted average of the motions that would result from rigid links to N independent degrees of freedom.

If we denote the six equations of rigid body motion in Equation 5-10 as:

$$\boldsymbol{u}_A = \boldsymbol{T}_A^T \boldsymbol{R}_{AB} \boldsymbol{T}_B \boldsymbol{u}_B \tag{5-12}$$

Then, using a least squares fit of the relationships of the RBE3, one obtains:

$$\boldsymbol{u}_D = \boldsymbol{T}_D^T \boldsymbol{A}^{-1} \sum_{i=1}^{N} \left[\boldsymbol{T}_i \boldsymbol{R}_{D_i} \right]^T \boldsymbol{w}_i \boldsymbol{u}_i \tag{5-13}$$

where $\boldsymbol{w}_i$ denotes a user-specified weighting matrix to alter the default curve fit and A is given by:

$$\boldsymbol{A} = \sum_{i=1}^{N} \left[\boldsymbol{T}_i \boldsymbol{R}_{D_i} \right]^T \boldsymbol{w}_i \left[\boldsymbol{T}_i \boldsymbol{R}_{D_i} \right] \tag{5-14}$$

5.2. APPLIED LOADS

Three types of loads can be applied in a static analysis:

- Mechanical or concentrated external loads
- Gravity loads
- Thermal loads

These load types may be applied separately or in combination. The last two load types have the potential to vary with the structural design and this fact is recognized in the generation of these loads. Each of the load types briefly discussed in the following sections.

5.2.1. Mechanical Loads

External loads are applied to the structural model in ASTROS through the use of input entries which define forces, moments and pressure loadings. The forces are applied at specified grid points and in a direction either defined explicitly in the input or by reference to two grid points which define a direction along which the force acts. Similarly, moments are applied at specified grid points and in a direction either defined explicitly in the input or by reference to two grid points which define an axis about which the load is applied.

Pressure loads are defined by specifying a pressure and an area over which it acts. The area is specified by reference to three or four grid points. In the case of three grids, the area of the resulting triangle is computed and the resulting force is distributed equally to the three grids. For the case of four grids, the surface is defined by two sets of overlapping triangles, half the pressure is applied to each set and the triangle algorithm is then applied. Input data descriptions for the **FORCE, MOMENT, FORCE1, MOMENT1,** and **PLOAD** Bulk Data entries in the ASTROS User's Manual entries contain further information on the preparation of this static loads data.

5.2.2. Gravity Loads

The gravity load is specified by a user defined acceleration and a direction. This acceleration vector is then applied to each grid point's translational degrees of freedom to obtain a global acceleration vector. No rotational accelerations are applied. The gravity loads are then computed by multiplying the mass matrix by this acceleration vector. Subsection 5.4 discusses the special treatment of gravity loads when the mass matrix is a function of the design variables.

5.2.3. Thermal Loads

A basic capability to consider thermal effects has been implemented in ASTROS through the specification of temperatures at grid points. For computing the thermal loads, this temperature is differenced from a reference temperature that is specified by the user for each material that is used in the structure.

For each finite element, a thermal load sensitivity vector is generated, as discussed in Subsection 5.1. If the element is designed, this vector is computed for the fixed value of the local design variable. Subsection 5.4 discusses the assembly of these thermal load components into a global load vector.

5.3. STRENGTH CONSTRAINTS

As discussed in Subsection 2.2.2.1, ASTROS supports four basic forms of element dependent strength constraints

- Von Mises failure criterion
- Tsai-Wu failure criterion
- Principal strain criterion
- Fiber/transverse strain criterion

The structural element design constraints are shown in Table 8. The Tsai-Wu constraint is not available for the one-dimensional elements and the shear panel since these elements support only isotropic material properties. The solid elements (IHEXi) may not be constrained in the design task and the stress constraint for the scalar spring element is imposed as a displacement constraint. The principal strain constraint generates two constraints for each element or composite laminate, one for each principal strain value. All other constraints generate one constraint per finite element, layer of a composite element, or stress computation point within an element.

Just as in the case of stiffness and mass matrices, it is desirable to compute the design invariant terms useful in stress/strain computations in order to speed processing within the design iteration loops. In ASTROS, this takes place in the **EMG** module in which the matrix S, that relates stress or strain components for each constrained element to the global displacements, is formed. (Note that it is not necessary to design an element in order to impose a stress or strain constraint.)

The S matrix is very similar to that formed in MSC/NASTRAN for the matrix method of stress recovery in dynamic response analyses (see Subsection 4.7 of Reference 15). It is used both to evaluate the strength constraints and to evaluate the constraint sensitivities to the global displacements according to the following expression:

$$\sigma = S^T u_g \tag{5-15}$$

where σ represents the element stress or strain components which are then combined to compute the particular strength constraint.

The static stress or strain may be expressed as a function of the design variables and of the static response (refer to Chapter 6 for the details):

$$\sigma = f(u, v) \tag{5-16}$$

Table 8. Structural Element Design Constraints

ELEMENT	ALLOWED STRENGTH CONSTRAINTS
BAR	von Mises
QDMEM1	all forms
QUAD4	all forms
ROD	von Mises, Principal Strain
SHEAR	von Mises, Principal Strain
TRMEM	all forms
TRIA3	all forms

The sensitivity of the j^{th} stress component to a change in the i^{th} design variable is given by:

$$\frac{\partial \sigma_j}{\partial v_i} = \frac{\partial f_j}{\partial v_i} + \frac{\partial f_j^T}{\partial u} \frac{\partial u}{\partial v_i} \tag{5-17}$$

For stress and strain constraints:

$$\frac{\partial f_j^T}{\partial u} = \boldsymbol{S} \tag{5-18}$$

Thus, it is clear that $\boldsymbol{S}$ plays an important role in sensitivity analysis as well as constraint evaluation. The computational efficiency that is gained by using the $\boldsymbol{S}$ terms in constraint evaluation and sensitivity evaluation is that 1) $\boldsymbol{S}$ is computed only once for each constrained element and 2) for linearly linked elements, $\boldsymbol{S}$ is design invariant and so only needs to be computed once. For nonlinear design variables, computational efficiency is retained in that $\boldsymbol{S}$ is recomputed only for nonlinearly linked elements. The first term in Equation 5-17, which is

$$\frac{\partial \boldsymbol{S}}{\partial v_i} \tag{5-19}$$

is also nonzero and must be computed by finite difference.

The following subsections present the details of the $\boldsymbol{S}$ calculations for each of the structural elements.

5.3.1. BAR Element

The bar element stress constraint matrix calculations are performed much like those in the standard element data recovery as shown in Subsection 8.2 of Reference 3. The only difference is that the combinatorial operations relating the element static forces to the stresses are performed on the matrices relating the forces to the displacements rather than on the forces themselves. The 6 x 1 vector of element forces, $\boldsymbol{P}$, is related to the displacements by:

$$\boldsymbol{P} = \boldsymbol{S}_a \, \boldsymbol{u}_a + \boldsymbol{S}_b \, \boldsymbol{u}_b \tag{5-20}$$

where a and b denote ends A and B of the bar element, respectively. Merging this expression to avoid distinguishing between nodal displacement vectors gives:

$$\boldsymbol{P} = \left[\boldsymbol{S}_a \mid \boldsymbol{S}_b \right] \left\{ \frac{\boldsymbol{u}_a}{\boldsymbol{u}_b} \right\} = \boldsymbol{S} \, \boldsymbol{u}_g \tag{5-21}$$

The 6 x 12 matrix $\boldsymbol{S}$ is computed from the element stiffness matrix as shown in Reference 3. Note that the effects of thermal loads are omitted from Equation 5-17. Unlike all other elements in ASTROS, the stress contribution due to thermal loads is design dependent for the bar element. This feature of the bar element is not supported in ASTROS for the linear linking option with the result that *linearized* design optimization with constrained bar elements under thermal loads is inaccurate. The resultant stress

constraints and constraint sensitivities are self-consistent but neither account for the stress-free strain arising from the thermal load. For nonlinear linking, these effects are handled correctly.

The bar element stresses are normally computed through combinations of the components of $\boldsymbol{P}$, the user input stress computation points C, D, E, and F, the moments of inertia, I_1, I_2, and I_{12}, and the bar element length. In order to form the $\boldsymbol{S}$ matrix, these linear combinations are instead performed on the rows of the matrix $\boldsymbol{S}$. Using the notation $\boldsymbol{S}_i$ to denote the i^{th} row of $\boldsymbol{S}$, the $\boldsymbol{S}$ columns relating to the stress components at ends A and B for the computation point defined by the user input points $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ are:

$$\frac{\partial \sigma_{ca}}{\partial \boldsymbol{u}} = \frac{C_1 I_{12} - C_2 I_1}{I_1 I_2 - I_{12}^2} S_5 + \frac{C_1 I_2 - C_2 I_{12}}{I_1 I_2 - I_{12}^2} S_6 \tag{5-22}$$

$$\frac{\partial \sigma_{ca}}{\partial \boldsymbol{u}} = \frac{C_1 I_{12} - C_2 I_1}{I_1 I_2 - I_{12}^2} \left[S_5 - l\left(S_3\right) \right] + \frac{C_1 I_2 - C_2 I_{12}}{I_1 I_2 - I_{12}^2} \left[S_6 - l\left(S_2\right) \right] \tag{5-23}$$

The remaining six stress components are computed in a similar manner.

In evaluating the stress constraints, the columns of $\boldsymbol{S}$ are multiplied by the displacement vectors to obtain the stress components. For the bar, each component then generates a single von Mises stress constraint. It is important to note that each bar element generates eight stress constraints for every load condition and that any coincident stress computation points will generate redundant constraints.

5.3.2. QDMEM1 Element

The isoparametric quadrilateral membrane element stress/strain constraint matrix calculations are performed much like those in the standard element data recovery as shown in Subsection 8.19 of Reference 3. Four 3 x 3 matrices that relate stresses to the individual nodal displacements are computed as:

$$\boldsymbol{S}_i = \left[\boldsymbol{GAB}^T \boldsymbol{E}^T \right]_i \boldsymbol{T}_i \quad ; i = 1, 2, 3, 4 \tag{5-24}$$

where $\boldsymbol{G}$ is the 3 x 3 stress-strain matrix, $\boldsymbol{A}$ is the 3 x 8 strain-displacement matrix evaluated at the intersection of the element diagonals, $\boldsymbol{B}$ is the 8 x 12 matrix relating nodal displacements to displacements in the element mean plane, $\boldsymbol{E}$ is the 12 x 12 matrix relating nodal displacements in the basic coordinate system to element coordinates, and $\boldsymbol{T}$ is the appropriate 3 x 3 transformation matrix from basic to global coordinates. The subscript i in Equation 5-24 denotes the appropriate matrix or matrix partition for the i^{th} node. The three stress components for the QDMEM1 element may be computed (neglecting thermal strains) from the matrices $\boldsymbol{S}$

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}_{MECH} = \sum_{i=1}^{4} \boldsymbol{S}_i \boldsymbol{u}_{gi} \tag{5-25}$$

which more clearly shows that $\boldsymbol{S}$ is formed directly from the rows of $\boldsymbol{S}$ if it is rewritten as:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}_{MECH} = \left[\boldsymbol{S}_1 \mid \boldsymbol{S}_2 \mid \boldsymbol{S}_3 \mid \boldsymbol{S}_4 \right] \begin{Bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \boldsymbol{u}_3 \\ \boldsymbol{u}_4 \end{Bmatrix} = \boldsymbol{S}^T \boldsymbol{u}_g \tag{5-26}$$

The product of $\boldsymbol{S}$ and the global displacements yield, for the QDMEM1 element, the three stress components in the element coordinate system. If thermal loads are applied, these components must be decremented by the amount of stress arising from the thermal strains. This is accomplished by separately storing the "thermal stress sensitivity" $\boldsymbol{S}_t$ vector for the element:

$$\boldsymbol{S}_t = \boldsymbol{G}\,\alpha \tag{5-27}$$

where α is the 3 x 1 vector of thermal expansion coefficients. This vector is then used in the stress constraint evaluation to compute the actual stress components as:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}_{TOT} = \boldsymbol{S}^T \boldsymbol{u}_g - \boldsymbol{S}_t \left[\boldsymbol{T} - \boldsymbol{T}_o \right] \tag{5-28}$$

The stress components are then used to evaluate the von Mises or Tsai-Wu stress constraints. The columns of $\boldsymbol{S}$ are also used to compute the stress constraint sensitivities. The thermal stress terms contribute only to the constraint evaluation and not to the constraint sensitivity.

For principal strain constraints, the operations of Equations 5-24 through 5-26 are carried out in an identical manner except that the stress-strain matrix $\boldsymbol{G}$ is omitted from Equation 5-24. This results in the computation of the three strain components for the element rather than the stress components. There is no correction required for thermal loads since the thermal strains are included in the calculation of the constraint.

5.3.3. QUAD4 Element

The quadrilateral plate bending element stress/strain constraint matrix is formed from the stress-strain and/or strain-displacement and the appropriate coordinate system transformation matrices presented in Appendix A. The three components of stress or strain in the element coordinate system at

the origin of the element coordinate system at the user specified fiber distances are thus related to the nodal displacements. These terms form the columns of the S matrix.

If thermal loads are applied, however, the stress components must be decremented by the amount of stress arising from the thermal strains. This is accomplished by separately storing the "thermal stress sensitivity" S_t vector for the element:

$$S_t = G\alpha \tag{5-29}$$

where α is the 3 x 1 vector of thermal expansion coefficients.

The stress or strain components can then be computed exactly as they are for the QDMEM1 element, with the exception that there are two sets of components for each element (one for each fiber distance). Those components are then used to evaluate the von Mises, Tsai-Wu or Principal Strain constraint.

For designed composite laminates, each ply is treated as a separate (membrane only) element with the result that each layer is treated exactly like a QDMEM1 element. For other laminates, the stress or strain constraint is applied to the element using the equivalent laminate properties and so is treated exactly as are metallic QUAD4 elements.

5.3.4. ROD Element

The rod element stress/strain constraint matrix calculations are performed much like those in the standard element data recovery as shown in Subsection 8.27 of Reference 3. The two sets of 3 x 3 matrices relating stresses to the individual nodal translations and rotations are computed exactly as shown in the reference giving four matrices:

S_a^t Stress-displacement matrix for translations at end A

S_b^t Stress-displacement matrix for translations at end B

S_a^r Stress-displacement matrix for rotations at end A

S_b^r Stress-displacement matrix for rotations at end B

The tensile and torsional stress constraint sensitivity components for the rod element are then formed as

$$\sigma_{MECH} = \left[S_a^t \mid S_b^t \right] \left\{ \frac{u_a^t}{u_b^t} \right\} \tag{5-30}$$

$$\tau_{MECH} = \left[S_a^r \mid S_b^r \right] \left\{ \frac{u_a^r}{u_b^r} \right\} \tag{5-31}$$

which show $\boldsymbol{S}$ to be formed directly from the rows of $\boldsymbol{S}$. In the case of thermal loads, the tensile stress values computed from the product of $\boldsymbol{S}$ and the global displacements must be decremented by the amount of stress arising from the thermal strain. This is accomplished by separately storing the "thermal stress sensitivity" $\boldsymbol{S}_t$ vector for the element:

$$\boldsymbol{S}_t = \alpha \boldsymbol{E} \tag{5-32}$$

where $\boldsymbol{E}$ is the Young's Modulus for the material and α is the thermal expansion coefficient. This vector is then used in the stress constraint evaluation to compute the actual stress component as:

$$\sigma_{TOT} = \boldsymbol{S}^T \boldsymbol{u}_g - \alpha E (T - T_o) \tag{5-33}$$

Both the tensile and torsional components are used to evaluate the von Mises stress constraints. The columns of $\boldsymbol{S}$ are also used to compute the stress constraint sensitivities. The thermal stress terms contribute only to the tensile stress component in the constraint evaluation and not to the constraint sensitivity.

For principal strain constraints, the operations of Reference 3 that generate the $\boldsymbol{S}$ matrices are modified to compute the strains rather than the stress components. The tensile and torsional strain components are used to compute the two principal strain values with no correction for thermal loads since the thermal strains are included in the calculation of the strain constraints.

5.3.5. SHEAR Panel

The shear panel stress/strain constraint matrix calculations are performed much like those in the standard element data recovery as shown in Subsection 8.3 of Reference 3. The average stress along the first side of the shear panel is computed as:

$$S_A = \sum_{i=1}^{4} \boldsymbol{S}_i \boldsymbol{u}_g^t \tag{5-34}$$

where the $\boldsymbol{S}_i$ are the stress/strain displacement matrices for each node as shown in Reference 3 and $\boldsymbol{u}_i^t$ are the nodal translations in global coordinates. From S_A, the corner stresses are computed based on four scalar coefficients $\boldsymbol{P}_i$ whose values are computed to account for parallelogram, trapezoid or general quadrilateral geometries. The average shear stress or strain, which is used in ASTROS for the constraint evaluation, is then computed as the average of the four corner stress/strain values. In order to compute the $\boldsymbol{S}$ terms, the corner stress calculations and averaging operation were merged with the $\boldsymbol{S}_A$ computations to give:

$$\sigma = \frac{1}{4}\left(\frac{\boldsymbol{P}_2}{\boldsymbol{P}_1} + \frac{\boldsymbol{P}_1}{\boldsymbol{P}_2} + \frac{\boldsymbol{P}_1 \boldsymbol{P}_2}{\boldsymbol{P}_3^2} + \frac{\boldsymbol{P}_1 \boldsymbol{P}_2}{\boldsymbol{P}_4^2}\right)\left[\boldsymbol{S}_1 \mid \boldsymbol{S}_2 \mid \boldsymbol{S}_3 \mid \boldsymbol{S}_4\right]\begin{Bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \boldsymbol{u}_3 \\ \boldsymbol{u}_4 \end{Bmatrix} \tag{5-35}$$

The product of $\boldsymbol{S}$ and the global displacements yield, for the SHEAR element, the average stress or strain for the shear panel. The shear panel does not support any thermal strains so no corrections are needed to the stress value. The average stress or strain is then used to evaluate the von Mises or Principal Strain constraints.

5.3.6. TRMEM Element

The constant strain triangular membrane element stress/strain constraint matrix calculations are performed much like those in the standard element data recovery as shown in Subsection 8.4 of Reference 3. The three 3 x 3 matrices relating stresses to the individual nodal displacements are computed as:

$$\boldsymbol{S}_i = \boldsymbol{G}\,\boldsymbol{C}_i\,\boldsymbol{E}^T\,\boldsymbol{T}_i \quad ; i = 1, 2, 3 \tag{5-36}$$

where $\boldsymbol{G}$ is the 3 x 3 stress-strain matrix, $\boldsymbol{C}$ is the appropriate 3 x 2 strain-displacement matrix, $\boldsymbol{E}$ is the 3 x 2 matrix relating nodal displacements in the basic coordinate system to element coordinates, and $\boldsymbol{T}$ is the appropriate 3 x 3 transformation matrix from basic to global coordinates. The subscript i in Equation 5-36 denotes the appropriate matrix or matrix partition for the i^{th} node. The three stress components for the TRMEM element may be computed (neglecting thermal strains) from the matrices $\boldsymbol{S}_i$ as

$$\left\{ \begin{matrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{matrix} \right\}_{MECH} = \sum_{i=1}^{3} \boldsymbol{S}_i\,\boldsymbol{u}_{gi} \tag{5-37}$$

which more clearly shows that $\boldsymbol{S}$ is formed directly from the rows of $\boldsymbol{S}_i$ if it is rewritten as:

$$\left\{ \begin{matrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{matrix} \right\}_{MECH} = \left[\boldsymbol{S}_1 \mid \boldsymbol{S}_2 \mid \boldsymbol{S}_3 \right] \left\{ \begin{matrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \boldsymbol{u}_3 \end{matrix} \right\} \tag{5-38}$$

The product of $\boldsymbol{S}$ and the global displacements yield, for the TRMEM element, the three stress components in the element coordinate system. If thermal loads are applied, these components must be decremented by the amount of stress arising from the thermal strains. This is accomplished by separately storing the "thermal stress sensitivity" $\boldsymbol{S}_t$ vector for the element:

$$\boldsymbol{S}_t = \boldsymbol{G}\,\alpha \tag{5-39}$$

where α is the 3 x 1 vector of thermal expansion coefficients. This vector is then used in the stress constraint evaluation to compute the actual stress components as:

$$\left\{ \begin{matrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{matrix} \right\}_{TOT} = \boldsymbol{S}^T\,\boldsymbol{u}_g - \boldsymbol{S}_t \left(T - T_o \right) \tag{5-40}$$

The stress components are then used to evaluate the von Mises or Tsai-Wu stress constraints. The columns of $\boldsymbol{S}$ are also used to compute the stress constraint sensitivities. The thermal stress terms contribute only to the constraint evaluation and not to the constraint sensitivity.

For principal strain constraints, the operations represented by Equations 5-36 through 5-38 are carried out in an identical manner except that the stress-strain matrix $\boldsymbol{G}$ is omitted from Equation 5-36. This results in the computation of the three strain components for the element rather than the stress components. There is no correction required for thermal loads since the thermal strains are included in the calculation of the constraint.

5.4. GLOBAL ASSEMBLY OF MATRICES

This section describes the assembly of the global mass, stiffness and applied loads matrices. The automated design capability in ASTROS makes it desirable to perform this assembly in two stages. In the first stage, matrices are assembled that are invariant with respect to the global design variables. In the second stage, these invariant matrices are multiplied by the current values of the global design variables, and combined with finite difference sensitivities for nonlinear design variables, to give the final matrices. Mathematically, for the stiffness matrix, the first stage entails forming a stiffness design sensitivity matrix of the form:

$$\frac{\partial \boldsymbol{K}_{gg}}{\partial v_i} = \mathbf{A} \sum_{i=1}^{nle} p_{ij}\, \boldsymbol{k}^{i}_{fact} + \mathbf{A} \sum_{i=1}^{nle} p_{ij}\, \frac{\partial \boldsymbol{k}_{ee}}{\partial t_i} \tag{5-41}$$

where $\frac{\partial \boldsymbol{K}_{gg}}{\partial v_i}$ is the stiffness design sensitivity matrix, P_{ij} is the scalar linking factor defined in Equation 4-6, $\boldsymbol{k}_{fact}$ are the factored element stiffness matrices associated with linear design variables, $\frac{\partial \boldsymbol{k}_{ee}}{\partial t_i}$ are the finite difference derivatives associated with nonlinear design variables, i is the subscript for the i^{th} local design variable, j is the subscript for the j^{th} global design variable, and *nle* is the number of local variables linked to the global variable.

The $\frac{\partial \boldsymbol{K}_{gg}}{\partial v_i}$ are global matrices and have rows and columns equal in number to the degrees of freedom in the *g-set* and are, therefore, potentially large, sparse matrices. These matrices are stored in ASTROS as unstructured entities with an associated relation providing information that identifies the degrees of freedom with which the global design variable is associated.

An equation similar to that of Equation 5-41 is used for the mass matrix:

$$\frac{\partial \boldsymbol{M}_{gg}}{\partial v_i} = \mathbf{A} \sum_{i=1}^{nle} p_{ij}\, \boldsymbol{m}^{i}_{fact} + \mathbf{A} \sum_{i=1}^{nle} p_{ij}\, \frac{\partial \boldsymbol{m}_{ee}}{\partial t_i} \tag{5-42}$$

where $\frac{\partial M_{gg}}{\partial v_i}$ is the mass design sensitivity matrix, m_{fact} are the factored element mass matrices associated with linear design variables, and $\frac{\partial m_{ee}}{\partial t_i}$ are the finite difference derivatives associated with nonlinear design variables.

Since, for linear design variables, the $\frac{\partial K_{gg}}{\partial v_i}$ and $\frac{\partial M_{gg}}{\partial v_i}$ matrices are independent of the values of the global design variables, the first term of the assembly process indicated in Equations 5-41 and 5-42 needs to be performed only once for a given design task. Another motivation for forming these matrices is that they are required in the sensitivity calculations.

Inside the design loop, the nonlinear elemental matrices are recomputed and a second assembly takes place to form the final global matrices:

$$K_{gg} = \mathbf{A}\, k^{ee}_{fixed} + \mathbf{A} \sum_j \sum_i p_{ij} k^{i}_{fact} v_j + \mathbf{A}\, k^{ee}_{nl} + \mathbf{A} \sum_j v_j^{\alpha} DKBV \tag{5-43}$$

$$M_{gg} = \mathbf{A}\, m^{ee}_{fixed} + \mathbf{A} \sum_j \sum_i p_{ij} m^{i}_{fact} v_j + \mathbf{A}\, m^{ee}_{nl} + \mathbf{A} \sum_j v_j^{\alpha} DKBV \tag{5-44}$$

where the ***M, K***, and ***v*** terms are defined in Subsection 4.3, *ndv* is the number of global design variables and *i* and *j* identify the design variable. The ***DKBV*** term of Equation 5-43 corresponds to the special case of bar elements as described in Subsection 5.1.3.2. The Equation 5-4 relation, in particular, indicates the source of this term and

$$DKBV_i = \sum_{j=1}^{nke} P_{ij}\, k^{R}_{ee_j} \tag{5-45}$$

If there are no bending effects, or if the BAR's are designed using fully nonlinear variables, this ***DKBV*** term is, of course, not present.

The global stiffness and mass matrices are typically sparse and strongly banded; i.e., the nonzero terms are located close to the matrix diagonal. These facts are utilized by both the data base in its handling of sparse matrices and by the large matrix utilities when these matrices, and their partitioned forms, undergo addition, multiplication, decomposition, etc.

The assembly of the global loads matrix takes a similar path. Outside the design loop, design invariant portions of the loads are assembled once as part of the preface operations. For the mechanical loads, there is no design dependent portion so that the entire assembly process essentially takes place at this time. The one exception to this is that ASTROS retains the NASTRAN concept of *simple* and *combined* loads that permit the user to specify a total loading condition that is the sum of several load vectors:

$$\boldsymbol{P}_g = S_0 \sum_i S_i \boldsymbol{L}_i \tag{5-46}$$

where $\boldsymbol{P}$ is the total load vector, $\boldsymbol{S}_0$ and $\boldsymbol{S}_i$ are scalar multipliers and $\boldsymbol{L}$ is a simple load. If this summation is required, it is performed inside the design loop to accommodate the possibility that a simple load may be required in more than one $\boldsymbol{P}$ vector.

The gravity and thermal loads are clearly design dependent. The gravity loads are simply:

$$\boldsymbol{P}_g^{GRAV} = \boldsymbol{M}_{gg}\, \boldsymbol{a}_g \tag{5-47}$$

where $\boldsymbol{a}_g$ is the global applied acceleration field. In ASTROS, $\boldsymbol{P}_g^{GRAV}$ is assembled from the elemental mass matrices from direct multiplication:

$$\boldsymbol{P}_g^{GRAV} = \mathbf{A}\, \boldsymbol{m}_{fixed}^{ee}\, \boldsymbol{a}_g^e + \mathbf{A} \sum_j \sum_i p_{ij}\, \boldsymbol{m}_{fact}^i \boldsymbol{a}_g^e v_j + \mathbf{A}\, \boldsymbol{m}_{nl}^{ee}\, \boldsymbol{a}_g^e \tag{5-48}$$

where the first two terms are computed and assembled once in the preface. The simple load sensitivities are readily computed as:

$$\frac{\partial \boldsymbol{P}_g^{GRAV}}{\partial v_j} = \mathbf{A} \sum_j \sum_i p_{ij}\, \boldsymbol{m}_{fact}^i \boldsymbol{a}_g^e + \mathbf{A} \sum_j \sum_i p_{ij} \frac{\partial \boldsymbol{m}_{ee}}{\partial t_i} \boldsymbol{a}_g^e \tag{5-49}$$

where again the first term is formed only once.

The global thermal load sensitivity vectors are a somewhat complicated combination of the fixed, factored, and nonlinear elemental thermal vectors, $\boldsymbol{T}_{ee}$, the grid point temperatures, T_{GRID}, and the material reference temperatures, T_{REF}.

$$\begin{aligned} \boldsymbol{P}_g^{THERM} = {} & \mathbf{A}\, \boldsymbol{T}_{fixed}^{ee} (T_{GRID} - T_{ref}) \\ & + \mathbf{A} \sum_j \sum_i p_{ij}\, \boldsymbol{T}_{fact}^i (T_{GRID} - T_{ref}) \\ & + \mathbf{A}\, \boldsymbol{T}_{nl}^{ee} (T_{GRID} - T_{ref}) \end{aligned} \tag{5-50}$$

where the first two terms are computed once and augmented in the design loop with the third term.

Similarly, the sensitivities are formed as:

$$\frac{\partial \boldsymbol{P}_g^{THERM}}{\partial v_j} = \mathbf{A} \sum_i p_{ij} \boldsymbol{T}_{fact}^i \left(T_{GRID} - T_{ref} \right) + \mathbf{A} \sum_i p_{ij} \frac{\partial \boldsymbol{T}^{ee}}{\partial t_i} \left(T_{GRID} - T_{ref} \right) \tag{5-51}$$

where only the second term is recomputed within the design loop.

6. STATIC ANALYSIS

The static analysis capability in ASTROS provides the capability to analyze and design linear structures subjected to time invariant loading. This section emphasizes the matrix algebra that is used in this analysis. This algebra is straightforward and should be familiar to most analysts, but it is described in some detail here since it is basic to the operation of the procedure, particularly as it applies to the standard MAPOL sequence described in Appendix C of the User's Manual. The presentation given here includes inertia relief terms throughout, even though this is a somewhat esoteric concept in structural analysis. It is included both because it provides the most general formulation and because it foreshadows the discussion of static aeroelasticity where inertia relief is central to the discussion of free flying aircraft. The notation of Subsection 2.3 is used extensively in this discussion and only the terms which have not been previously defined are defined here.

6.1. MATRIX EQUATIONS FOR STATIC ANALYSIS

The equilibrium equation for ASTROS static analysis in the *g-set* is:

$$\boldsymbol{K}_{gg}\,\boldsymbol{u}_g + \boldsymbol{M}_{gg}\,\ddot{\boldsymbol{u}}_g = \boldsymbol{P}_g \tag{6-1}$$

Following the hierarchy of Figure 8, the *g-set* is partitioned into *m-set* and *n-set*. The relationship between these dependent and independent degrees of freedom is given by matrix $\boldsymbol{T}_{mn}$:

$$\boldsymbol{u}_m = \boldsymbol{T}_{mn}\,\boldsymbol{u}_n \tag{6-2}$$

An identical relation holds for the accelerations.

These multipoint constraints produce forces on the structure which are designated $\boldsymbol{c}_g$. The work performed by these forces must be, by definition, equal to zero. Subsection 5.4 of Reference 1 demonstrates that this work consideration leads to a condition that the constraint forces have an equation similar to Equation 6-2:

$$\boldsymbol{c}_g = \begin{bmatrix} -\boldsymbol{T}_{mn}^T \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{q}_m \equiv 0 \tag{6-3}$$

where the $\boldsymbol{q}_m$ are unknown forces that are included in the solution process. Equations 6-1, 6-2 and 6-3 can be combined to give :

$$\begin{bmatrix} \overline{K}_{nn} & K_{nm} & T_{mn}^T \\ K_{mn} & K_{mn} & -I \\ T_{mn} & -I & 0 \end{bmatrix} \begin{Bmatrix} u_n \\ u_m \\ q_m \end{Bmatrix} + \begin{bmatrix} \overline{M}_{nn} & M_{nm} \\ M_{mn} & M_{mm} \\ T_{mn} & -I \end{bmatrix} \begin{Bmatrix} \ddot{u}_n \\ \ddot{u}_m \end{Bmatrix} = \begin{Bmatrix} \overline{P}_n \\ P_m \\ 0 \end{Bmatrix}$$

where the bar over certain terms refers to the elements in the partitions of g-size matrix before reduction to the *n-set*. This notation is used throughout this section. These equations can be solved for u_n and $\ddot{u}_n$ in terms of q_m, u_m, and u_m to give:

$$K_{nn} u_n + M_{nn} \ddot{u}_n = P_n$$

where:

$$\begin{aligned} K_{nn} &= \overline{K}_{nn} + K_{nm} T_{mn} + T_{mn}^T \left[K_{mn} + K_{mm} T_{mn} \right] \\ M_{nn} &= \overline{M}_{nn} + M_{nm} T_{mn} + T_{mn}^T \left[M_{mn} + M_{mm} T_{mn} \right] \end{aligned} \tag{6-4}$$

$$P_n = \overline{P}_n + T_{mn}^T P_m \tag{6-5}$$

The next set of reductions involve the forces of single-point constraint. These constraints are of the form:

$$u_s = Y_s \tag{6-6}$$

The accelerations associated with these degrees of freedom are zero.

If these constraints are placed in Equation 6-4, the partitioned equations are:

$$\begin{bmatrix} K_{ff} & K_{fs} \\ K_{sf} & K_{ss} \end{bmatrix} \begin{Bmatrix} u_f \\ Y_s \end{Bmatrix} + \begin{bmatrix} M_{ff} & M_{fs} \\ M_{sf} & M_{ss} \end{bmatrix} \begin{Bmatrix} \ddot{u}_f \\ 0 \end{Bmatrix} = \begin{Bmatrix} \overline{P}_f \\ P_s \end{Bmatrix} \tag{6-7}$$

and the reduction to the *f-set* is done by retaining the first row of Equation 6-7:

$$K_{ff} u_f + M_{ff} \ddot{u}_f = P_f \tag{6-8}$$

where

$$P_f = \overline{P}_f - K_{fs} Y_s \tag{6-9}$$

The reduction to the *a-set* involves further partitioning of Equation 6-8 to give :

$$\begin{bmatrix} K_{aa} & K_{ao} \\ K_{ao}^T & K_{oo} \end{bmatrix} \begin{Bmatrix} u_a \\ u_o \end{Bmatrix} + \begin{bmatrix} M_{aa} & M_{ao} \\ M_{ao}^T & M_{oo} \end{bmatrix} \begin{Bmatrix} \ddot{u}_a \\ \ddot{u}_o \end{Bmatrix} = \begin{Bmatrix} \overline{P}_a \\ P_o \end{Bmatrix} \tag{6-10}$$

In a manner consistent with Guyan reduction, the mass matrix is reduced using a static condensation transformation of the mass matrix to relate the omitted and retained degrees of freedom:

$$\ddot{\boldsymbol{u}}_o = -\left[\boldsymbol{K}_{oo}^{-1}\boldsymbol{K}_{oa}\right]\ddot{\boldsymbol{u}}_a = \boldsymbol{G}_o\,\ddot{\boldsymbol{u}}_a \tag{6-11}$$

The stiffness reduction is performed using the more exact form:

$$\boldsymbol{u}_o = \boldsymbol{K}_{oo}^{-1}\boldsymbol{P}_o - \boldsymbol{K}_{oo}^{-1}\boldsymbol{K}_{oa}\,\boldsymbol{u}_a \tag{6-12}$$

These reductions can be applied with Equation 6-10 to give:

$$\boldsymbol{K}_{aa}\,\boldsymbol{u}_a + \boldsymbol{M}_{aa}\,\ddot{\boldsymbol{u}}_a = \boldsymbol{P}_a \tag{6-13}$$

where

$$\begin{aligned}
\boldsymbol{K}_{aa} &= \overline{\boldsymbol{K}}_{aa} + \boldsymbol{K}_{ao}\,\boldsymbol{G}_o \\
\boldsymbol{P}_a &= \overline{\boldsymbol{P}}_a - \boldsymbol{G}_o^T\boldsymbol{P}_o \\
\boldsymbol{M}_{aa} &= \overline{\boldsymbol{M}}_{aa} + \boldsymbol{M}_{oa}\,\boldsymbol{G}_o + \boldsymbol{G}_o\,\boldsymbol{M}_{oa} + \boldsymbol{G}_o^T\boldsymbol{M}_{oo}\,\boldsymbol{G}_o
\end{aligned} \tag{6-14}$$

It must be emphasized that the Guyan reduction of Equation 6-13 is approximate in that deformations due to inertial forces applied to the omitted degrees of freedom are neglected. The specification of the *a-set* degrees of freedom is therefore critical and places a burden on the user to take care in this specification. The dynamic reduction technique, described in Subsection 7.1, provides an alternative that is less demanding of the user.

A final point on the reduction to the *a-set* is that the reduction of Equation 6-13 is exact if there are no mass terms. Therefore, if a modal analysis and a static analysis, without inertia relief, share a boundary condition, the static analysis results will be the same as if no reduction took place and the modal analysis can benefit from any efficiency considerations associated with a reduced size eigenanalysis.

With the matrices in the *a-set*, the final partition is to the *l-set* and *r-set*:

$$\begin{bmatrix} \boldsymbol{K}_{ll} & \boldsymbol{K}_{lr} \\ \boldsymbol{K}_{rl} & \boldsymbol{K}_{rr} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_l \\ \boldsymbol{u}_r \end{Bmatrix} + \begin{bmatrix} \boldsymbol{M}_{ll} & \boldsymbol{M}_{lr} \\ \boldsymbol{M}_{rl} & \boldsymbol{M}_{rr} \end{bmatrix} \begin{Bmatrix} \ddot{\boldsymbol{u}}_l \\ \ddot{\boldsymbol{u}}_r \end{Bmatrix} = \begin{Bmatrix} \overline{\boldsymbol{P}}_l \\ \boldsymbol{P}_r \end{Bmatrix} \tag{6-15}$$

The *r-set* contains degrees of freedom equal in number to the number of rigid body modes in the structure. In ASTROS, as in NASTRAN, the *r-set* displacements are arbitrarily set to zero. The ASTROS methodology for solving Equation 6-15 starts from a determination of the rigid body mode shapes. These shapes could be determined from a direct consideration of the geometry of the structure, but they are determined in NASTRAN and ASTROS by solving for the displacements of an unloaded structure using the stiffness matrices:

$$\boldsymbol{u}_l = -\boldsymbol{K}_{ll}^{-1}\boldsymbol{K}_{lr}\boldsymbol{u}_r \tag{6-16}$$

with

$$\boldsymbol{D} = -\boldsymbol{K}_{ll}^{-1}\boldsymbol{K}_{lr} \tag{6-17}$$

designated the rigid body transformation matrix. Since the accelerations include only rigid body motions, it is possible to specify a relationship for the accelerations of Equation 6-15:

$$\ddot{\boldsymbol{u}}_l = \boldsymbol{D}\ddot{\boldsymbol{u}}_r \tag{6-18}$$

If Equation 6-18 is substituted into Equation 6-15, the following relationship results:

$$\begin{bmatrix} \boldsymbol{K}_{ll} & \boldsymbol{K}_{lr} & \boldsymbol{M}_{ll}\boldsymbol{D}+\boldsymbol{M}_{lr} \\ \boldsymbol{K}_{rl} & \boldsymbol{K}_{rr} & \boldsymbol{M}_{rl}\boldsymbol{D}+\boldsymbol{M}_{rr} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_l \\ \boldsymbol{u}_r \\ \ddot{\boldsymbol{u}}_r \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{P}_l \\ \boldsymbol{P}_r \end{Bmatrix} \tag{6-19}$$

If the first row of this equation is multiplied by $\boldsymbol{D}^T$, and added to the second, a simplified form of Equation 6-19 results:

$$\begin{bmatrix} \boldsymbol{K}_{ll} & \boldsymbol{K}_{lr} & \boldsymbol{M}_{ll}\boldsymbol{D}+\boldsymbol{M}_{lr} \\ \boldsymbol{R}_{31} \equiv 0 & \boldsymbol{R}_{32} \equiv 0 & \boldsymbol{m}_r \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_l \\ \boldsymbol{u}_r \\ \ddot{\boldsymbol{u}}_r \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{P}_l \\ \boldsymbol{D}^T\boldsymbol{P}_l+\boldsymbol{P}_r \end{Bmatrix} \tag{6-20}$$

where

$$\boldsymbol{m}_r = \boldsymbol{D}^T\boldsymbol{M}_{ll}\boldsymbol{D} + \boldsymbol{D}^T\boldsymbol{M}_{lr} + \boldsymbol{M}_{rl}\boldsymbol{D} + \boldsymbol{M}_{rr} \tag{6-21}$$

is the rigid body mass matrix. The R_{31} term of the left-hand matrix of Equation 6-20 is zero based on the definition of the D matrix given in Equation 6-17. The R_{32} term, which is

$$\boldsymbol{D}^T\boldsymbol{K}_{lr} + \boldsymbol{K}_{rr} \tag{6-22}$$

is zero because it represents the work performed on the structure when it undergoes a rigid body displacement.

The third row of Equation 6-20 can be solved for the accelerations in the *r-set* and these can be substituted into the first row along with the restraint condition $\boldsymbol{u}_r \equiv 0$ to solve for $\boldsymbol{u}_l$, the elastic deformations relative to the support point. $\boldsymbol{u}_a$ can then be recovered as

$$\boldsymbol{u}_a = \begin{Bmatrix} \boldsymbol{u}_l \\ 0 \end{Bmatrix} \tag{6-23}$$

Equation 6-18 is used to recover the accelerations in the *l-set*, which are then merged with the *r-set* accelerations to give $\ddot{\boldsymbol{u}}_a$.

Before continuing the recovery process, it should be noted that the solution process when no inertia terms are included is simply (from Equation 6-13)

$$\boldsymbol{K}_{aa}\,\boldsymbol{u}_a = \boldsymbol{P}_a \tag{6-24}$$

and it is possible to solve for $\boldsymbol{u}_a$ directly.

Once the displacements and accelerations have been computed in the *a-set*, it is a simple matter to recover to the *g-set*. The *o-set* accelerations are recovered directly using Equation 6-11:

$$\ddot{\boldsymbol{u}}_o = \boldsymbol{G}_o\,\ddot{\boldsymbol{u}}_a \tag{6-25}$$

while the *o-set* displacements recovery first requires that the applied loads be modified to include the inertia effects:

$$\boldsymbol{P}_o^i = -\left[\boldsymbol{M}_{oo}\,\ddot{\boldsymbol{u}}_o + \boldsymbol{M}_{oa}\,\ddot{\boldsymbol{u}}_a\right] = \left[\boldsymbol{M}_{oo}\boldsymbol{G}_o + \boldsymbol{M}_{oa}\right]\,\ddot{\boldsymbol{u}}_a \tag{6-26}$$

Equation 6-12 then gives

$$\boldsymbol{u}_o = \boldsymbol{K}_{oo}^{-1}\left[\boldsymbol{P}_o + \boldsymbol{P}_o^i\right] + \boldsymbol{G}_o\,\boldsymbol{u}_a \tag{6-27}$$

Merging the *a-set* and *o-set* degrees of freedom results in *f-set* displacements and accelerations. The s-set accelerations are zero and the displacements are contained in the $\boldsymbol{Y}_s$ vector of Equation 6-6 so that recovery of *n-set* degrees of freedom is immediate. Finally, the *m-set* dependent displacements and accelerations are recovered using Equation 6-2 and those are merged with the *n-set* vectors to give the displacements in the *g-set*.

6.2. CONSTRAINT EVALUATION

Static analyses have the potential of producing displacement and strength constraints. Given the global displacement vector recovered in the previous subsection, it is possible to evaluate these constraints directly. Separate modules in ASTROS evaluate the two types of constraints. The displacement constraints can be evaluated directly using the definition given in Equations 4-21 and 4-22. Strength constraints are evaluated in a two step process wherein the stress (or strain) components are first obtained by performing the matrix multiply of Equation 5-15

$$\sigma = \boldsymbol{S}^T\,\boldsymbol{u}_g \tag{6-28}$$

and then the constraints themselves are computed, based on the constraint type and the element type, as discussed in Subsections 4.3 and 5.3. It can perhaps be appreciated that the majority of the effort involved in evaluating these constraints is of a bookkeeping nature.

6.3. SENSITIVITY ANALYSIS

The final portion of the static analysis is the determination of the sensitivity of the constraints to changes in the design variables. The static analysis constraints can be expressed as functions of the design variables and the static response:

$$g = f(u, v) \tag{6-29}$$

The sensitivity of the j^{th} constraint to a change in the i^{th} design variable is given by

$$\frac{\partial g_j}{\partial v_i} = \frac{\partial f_j}{\partial v_i} + \frac{\partial f_j^T}{\partial u} \frac{\partial \boldsymbol{u}}{\partial v_i} \tag{6-30}$$

In general, static constraints may be directly dependent on the design variable $\left(\frac{\partial f_j}{\partial v_i} \neq 0\right)$ or only indirectly dependent $\left(\frac{\partial f_j}{\partial v_i} \equiv 0\right)$. For the case of linear design variables ($\boldsymbol{K}$, $\boldsymbol{M}$, $\boldsymbol{B}$ are linear in v), $\frac{\partial f_j}{\partial v_i} \equiv 0$ for stress, strain and displacement constraints. For nonlinear design variables, however, the term $\frac{\partial f_j}{\partial v_i}$ is nonzero.

Returning to Equation 6-30, strength constraint sensitivities are evaluated using the first and second terms. The $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}$ portion is computed using straightforward chain rule operations. Calculation of this term for displacement constraints and for von Mises stress constraints are given here as examples that should be adequate for motivating how the term would be evaluated for the remaining constraints.

6.3.1. Displacement Constraints

Upper bound displacement constraints are defined in ASTROS as (see Equation 4-21).

$$g = \sum_{i=1}^{ndisp} \frac{a_i u_i}{\delta_{i_{all}}} - 1.0 \tag{6-31}$$

where $\delta_{i_{all}}$ is the allowable upper bound.

The $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}$ term is a vector that is computed in the global analysis set. The only nonzero terms in this vector are associated with the degrees of freedom of the displacements included in the constraint. These values are $\frac{a_i}{\delta_{i_{all}}}$. For displacements, $\frac{\partial f}{\partial v}$ is zero for linear and nonlinear variables.

6.3.2. Von Mises Stress Constraints

Von Mises constraints are defined in ASTROS as (see Equation 2-8)

$$g = \left[\left(\frac{\sigma_x}{S_1}\right)^2 + \left(\frac{\sigma_y}{S_2}\right)^2 - \frac{\sigma_x \sigma_y}{S_1 S_2} + \left(\frac{\tau_{xy}}{F_s}\right)^2 \right]^{\frac{1}{2}} - 1.0 \tag{4-8}$$

The $\frac{\partial f}{\partial u}$ term for this constraint is derived to be

$$\frac{\partial f}{\partial u} = \frac{1}{2\,(g+1.0)} \left[\left(\frac{2\sigma_x}{S_1^2} - \frac{\sigma_y}{S_1 S_2} \right) \frac{\partial \sigma_x}{\partial u} + \left(\frac{2\sigma_y}{S_2^2} - \frac{\sigma_x}{S_1 S_2} \right) \frac{\partial \sigma_y}{\partial u} + \frac{2\tau_{xy}}{F_s^2} \frac{\partial \tau_{xy}}{\partial u} \right] \tag{6-32a}$$

Equation 6-28 is used to supply the gradients of the stress components with respect to the displacements. They are columns of the $\boldsymbol{S}$ matrix. The overall $\frac{\partial f}{\partial u}$ vector is therefore the weighted sum of up to three columns in the $\boldsymbol{S}$ matrix depending on the terms used in the constraint. The $\frac{\partial f}{\partial v}$ term is computed for **nonlinear design variables only**. Using a finite difference approach, the direct sensitivity of the $\boldsymbol{S}$ matrix is computed in a similar manner, providing the $\frac{\partial f}{\partial v}$ term.

$$\frac{\partial f}{\partial v} = \frac{1}{2\,(g+1.0)} \left[\left(\frac{2\sigma_x}{S_1^2} - \frac{\sigma_y}{S_1 S_2} \right) \left(\frac{\partial \boldsymbol{S}_x}{\partial v} \boldsymbol{u} \right) + \left(\frac{2\sigma_y}{S_2^2} - \frac{\sigma_x}{S_1 S_2} \right) \left(\frac{\partial \boldsymbol{S}_y}{\partial v} \boldsymbol{u} \right) + \frac{2\tau_{xy}}{F_S^2} \left(\frac{\partial \boldsymbol{S}_{xy}}{\partial v} \boldsymbol{u} \right) \right] \tag{6-32b}$$

It would appear that the only remaining task to complete the sensitivity analysis is the computation of the $\frac{\partial u}{\partial v}$ vector. In many cases, this is true, but ASTROS also contains an alternative analysis procedure that does not require the explicit calculation of this vector. These two alternatives, designated the gradient and the virtual load methods, are now described in a qualitative manner. This is followed by a more detailed formulation of the methods as they are implemented in ASTROS. References 17 and 18 provide a more general formulation and discussion of the two methods.

The basic equation for static analysis is

$$\boldsymbol{K u} = \boldsymbol{P} \tag{6-33}$$

This equation is written without regard to displacement set, hence, its qualitative nature. The sensitivity of the displacements to a design variable can be written as

$$K\frac{\partial u}{\partial v} = \frac{\partial P}{\partial v} - \frac{\partial K}{\partial v}u \tag{6-34}$$

Note that Equations 6-33 and 6-34 have the same stiffness matrix on the left-hand side and this similarity is exploited in ASTROS by storing the decomposed stiffness matrix when it is computed during the solution of Equation 6-33 and then retrieving this matrix for the solution of Equation 6-34. This straightforward approach to obtaining $\frac{\partial u}{\partial v}$ is designated the gradient approach in ASTROS terminology.

The alternative, virtual loads method, solves for the virtual displacements that would result if the $\frac{\partial f}{\partial u}$ vector were applied as a load to the structure:

$$K w = \frac{\partial f}{\partial u} \tag{6-35}$$

where w is the virtual displacement and, again, the similarity of Equation 6-33 to Equation 6-35 is used to avoid unnecessary decompositions of the stiffness matrix. If Equations 6-30, 6-34 and 6-35 are combined, the constraint sensitivity can be written as

$$\frac{\partial g}{\partial v} = w^T\left[\frac{\partial P}{\partial v} - \frac{\partial K}{\partial v}u\right] + \frac{\partial f}{\partial v} \tag{6-36}$$

If inertia relief effects are included in the static analysis, the virtual load approach to sensitivity analysis does not apply since the $K^{-T}K = I$ simplification required in Equation 6-36 is no longer possible. The gradient approach is therefore always used for the somewhat esoteric task of designing a structure while including inertia relief effects. For the more typical static analysis without inertia relief, the standard MAPOL sequence selects the approach that requires the least number of forward-backward substitutions; i.e., whether Equation 6-34 or 6-35 has the fewer right-hand sides. For Equation 6-34, the number of right-hand sides is equal to the number of active load cases times the number of design variables. For Equation 6-35, the number of right-hand sides is equal to the number of active displacement dependent constraints. It is difficult to generalize as to which approach will be chosen in a typical, real world design task, but it should be obvious that, for a large problem, one method could be significantly more efficient than the other. The actual calculations used in ASTROS for these two approaches are now given.

6.3.3. The Gradient Method

As indicated above, the gradient method of sensitivity evaluation is a straightforward application of derivative operations. In ASTROS, the formulation starts from taking the derivative of Equation 6-1 with respect to a design variable:

$$K_{gg}\frac{\partial u_g}{\partial v} + M_{gg}\frac{\partial \ddot{u}_g}{\partial v} = \frac{\partial P_g}{\partial v} - \frac{\partial K_{gg}}{\partial v}u_g - \frac{\partial M_{gg}}{\partial v}\ddot{u}_g \tag{6-37}$$

This equation has been written with the known terms on the right-hand side and the unknowns are the sensitivities of the displacements and accelerations in the *g-set*. Equation 6-37 is solved by going through a reduction and recovery process much like that given in Subsection 6.2 for the solution of Equation 6-1. In fact, the left-hand side reductions of the mass and stiffness matrices are identical in the two solutions so that these reductions are not repeated here, nor are they repeated in the ASTROS procedure.

The first term on the right-hand side is the sensitivity of the applied loads to the design variables. Subsection 5.4 shows that only gravity and thermal loads can vary with the design and that the sensitivity of these loads to the i^{th} design variable is simply

$$\left\{\frac{\partial \boldsymbol{P}_g}{\partial v_i}\right\}_{GRAV} \quad \text{and} \quad \left\{\frac{\partial \boldsymbol{P}_g}{\partial v_i}\right\}_{THERM} \tag{6-38}$$

Similarly, the sensitivity of the stiffness matrix to the i^{th} design variable is, from Equation 5-43

$$\frac{\partial \boldsymbol{K}_{gg}}{\partial v_j} = \mathbf{A} \sum_i p_{ij} \boldsymbol{k}^i_{fact} + \mathbf{A} \sum_i p_{ij} \frac{\partial \boldsymbol{k}_{ee}}{\partial t_i} + \alpha v_i^{(\alpha-1)} \boldsymbol{DKBV}_i \tag{6-39}$$

where the third term is zero except for the special case for the linearized design of bars. The sensitivity of the mass matrix is, from Equation 5-44:

$$\frac{\partial \boldsymbol{M}_{gg}}{\partial v_j} = \mathbf{A} \sum_i p_{ij} \boldsymbol{m}^i_{fact} + \mathbf{A} \sum_i p_{ij} \frac{\partial \boldsymbol{m}_{ee}}{\partial t_i} \tag{6-40}$$

For ease of notation, the right hand side of Equation 6-37 is designated $\frac{\partial \boldsymbol{R}_g}{\partial v_i}$ in the following, where

$$\frac{\partial \boldsymbol{R}_g}{\partial v_i} = \frac{\partial \boldsymbol{P}^{GRAV}_g}{\partial v_i} + \frac{\partial \boldsymbol{P}^{THERM}_g}{\partial v_i} - \frac{\partial \boldsymbol{K}_{gg}}{\partial v_i} \boldsymbol{u}_g - \frac{\partial \boldsymbol{M}_{gg}}{\partial v_i} \ddot{\boldsymbol{u}}_g \tag{6-41}$$

The specification of this pseudo-load in other displacement sets then follows the previous convention of using $\frac{\partial \boldsymbol{R}}{\partial v}$ to indicate the vector and the subscript to indicate the set. The last term in Equation (6-41) is needed for the inertia relief formulation of Subsection 6.1.

The reduction of Equation (6-41) to the *n-set* follows that given for the applied loads in Subsection 6.1:

$$\frac{\partial \boldsymbol{R}_n}{\partial v_i} = \overline{\frac{\partial \boldsymbol{R}_n}{\partial v_i}} + \boldsymbol{T}_{mn} \frac{\partial \boldsymbol{R}_m}{\partial v_i} \tag{6-42}$$

The single point constraints are removed by a partition of the *n-set* vectors to give $\frac{\partial \boldsymbol{R}_f}{\partial v_i}$,while the omitted degrees of freedom contribute to the *a-set*:

$$\frac{\partial \boldsymbol{R}_a}{\partial v_i} = \overline{\frac{\partial \boldsymbol{R}_a}{\partial v_i}} + \boldsymbol{G}_o \frac{\partial \boldsymbol{R}_o}{\partial v_i} \tag{6-43}$$

These pseudo-load vectors can be further partitioned into the *l-set* and *r-set* and an equation equivalent to that of Equation 6-20 can be written:

$$\begin{bmatrix} \boldsymbol{K}_{ll} & \boldsymbol{K}_{lr} & \boldsymbol{M}_{ll}\boldsymbol{D} + \boldsymbol{M}_{lr} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{m}_r \end{bmatrix} \begin{Bmatrix} \frac{\partial \boldsymbol{u}_l}{\partial v_i} \\ \frac{\partial \boldsymbol{u}_r}{\partial v_i} \\ \frac{\partial \ddot{\boldsymbol{u}}_r}{\partial v_i} \end{Bmatrix}_i = \begin{Bmatrix} \frac{\partial \boldsymbol{R}_l}{\partial v} \\ \\ \boldsymbol{D}^T \frac{\partial \boldsymbol{R}_l}{\partial v_i} + \frac{\partial \boldsymbol{R}_r}{\partial v_i} \end{Bmatrix}_i \tag{6-44}$$

where the $\frac{\partial \boldsymbol{u}}{\partial v}$ vectors and their accompanying subscripts designate the sensitivity of the particular displacement set to the i^{th} design variable and the $\frac{\partial \ddot{\boldsymbol{u}}}{\partial v}$ vectors similarly designate the sensitivity of accelerations.

The second row of Equation 6-44 can now be solved for $\frac{\partial \ddot{\boldsymbol{u}}_r}{\partial v_i}$, the sensitivities of the accelerations in the *r-set*, and these can then be substituted into the first row along with the constraint relation $\frac{\partial \boldsymbol{u}_r}{\partial v} \equiv \boldsymbol{0}$ to solve directly for $\frac{\partial \boldsymbol{u}_l}{\partial v_i}$, the sensitivities of the elastic deformations in the *l-set*. $\frac{\partial \ddot{\boldsymbol{u}}_l}{\partial v_i}$ is (from Equation 6-18) equal to $\boldsymbol{D}\frac{\partial \ddot{\boldsymbol{u}}_r}{\partial v_i}$. Unlike the analysis equations, it is not necessary to further recover the accelerations since, as Equation 6-30 indicates, the constraint sensitivity information is only a function of the displacement sensitivities. Another subtle point is that the vector multiplication indicated by the second term in Equation 6-30 gives the same scalar result in the f displacement set as it does if the calculations are performed in the significantly larger *g-set*. A substantial efficiency can then result when it is considered that this vector multiplication (which can have hundreds to thousands of terms) is required for the sensitivity of all the active constraints with respect to all the design variables. Therefore, the displacement sensitivities are recovered only up to the *f-set*, which requires computing the sensitivities of the omitted degrees of freedom in a fashion similar to Equation 6-27.

$$\frac{\partial \boldsymbol{u}_o}{\partial v_i} = \boldsymbol{K}_{oo}^{-1} \left[\frac{\partial \boldsymbol{R}_o}{\partial v_i} - \left[\boldsymbol{M}_{oo}\boldsymbol{G}_o + \boldsymbol{M}_{oa} \right] \frac{\partial \boldsymbol{u}_a}{\partial v_i} \right] + \boldsymbol{G}_o \frac{\partial \boldsymbol{u}_a}{\partial v_i} \tag{6-45}$$

and $\frac{\partial \boldsymbol{u}_f}{\partial v_i}$ is obtained by merging the *o-set* and *a-set* vectors.

A remaining step is the reduction of the $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}$ vector to the *f-set*. This reduction also follows that of the applied loads so that the reductions are:

$$\frac{\partial f_j}{\partial \boldsymbol{u}_n} = \frac{\overline{\partial f_j}}{\partial \boldsymbol{u}_n} + \boldsymbol{T}_{mn} \frac{\partial f_j}{\partial \boldsymbol{u}_m} \tag{6-46}$$

and $\frac{\partial f_j}{\partial \boldsymbol{u}_f}$ is obtained from a simple partitioning operation.

All the terms are now in place to calculate the constraint sensitivity. The mechanics of this calculation are rather complex since, although this discussion has been in terms of calculating the sensitivity of a single constraint to a single design variable, the calculations are performed in ASTROS in a much more terse fashion. For example, the load sensitivity vectors for all the design variables are computed simultaneously so that the $\frac{\partial \boldsymbol{R}_g}{\partial v_i}$ vector becomes a matrix and the reduction and forward/backward substitution processes are matrix operations. Similarly, the $\frac{\partial f_j}{\partial \boldsymbol{u}_g}$ vectors for all the constraints are computed and reduced simultaneously. The matrix which gives the sensitivities of all the constraints to all the design variables is

$$\boldsymbol{A}^T = \frac{\partial f_j^T}{\partial \boldsymbol{u}_f} \frac{\partial \boldsymbol{u}_f}{\partial v} \tag{6-47}$$

where $\boldsymbol{A}$ has a row dimension equal to *ndv*, the number of global design variables and a column dimension equal to *nac*, the number of active constraints. $\frac{\partial f_j}{\partial \boldsymbol{u}_f}$ is of dimension f-size, the number of degrees of freedom in the *f-set*, by *nac*, and $\frac{\partial \boldsymbol{u}_f}{\partial v}$ is of dimension f-size by *ndv* times *nalc*. The *nalc* term is the number of active load cases. The matrix multiplication indicated in Equation 6-47 is not conformable when *nalc* is greater than one. It is therefore necessary to perform partitioning operations inside an ASTROS module to subdivide the matrices into the proper conforming form.

Note that the $\boldsymbol{A}$ matrix of Equation 6-47 contains only the constraints produced by the static analyses. Thickness, frequency and aeroelastic constraint sensitivities must be appended onto this matrix before the redesign process can take place.

6.3.4. The Virtual Displacement Method

As indicated by Equation 6-35, the virtual displacement method entails solving for right-hand side vectors that are based on the sensitivity of the constraint to the displacement. These vectors must be reduced to the *a-set* (recall that inertia relief is not supported for this option so that reduction beyond the *a-set* is not possible or necessary). The reduction to the *f-set* has already been described in Equation 6-46 and the accompanying text. The reduction from the *f-set* to the *a-set* is simply

$$\frac{\partial f_j}{\partial \boldsymbol{u}_a} = \frac{\overline{\partial f_j}}{\partial \boldsymbol{u}_a} + \boldsymbol{G}_o \frac{\partial f_j}{\partial \boldsymbol{u}_o} \tag{6-48}$$

Given these vectors, the virtual displacements are calculated from

$$\boldsymbol{K}_{aa} \boldsymbol{w}_{aj} = \frac{\partial f_j}{\partial \boldsymbol{u}_a} \tag{6-49}$$

The omitted virtual displacements are recovered using

$$\boldsymbol{w}_{oj} = \boldsymbol{K}_{oo}^{-1} \frac{\partial f_j}{\partial \boldsymbol{u}_o} + \boldsymbol{G}_o \boldsymbol{w}_{aj} \tag{6-50}$$

A merge operation produces $\boldsymbol{w}_{f_j}$ and Equation 6-36 is used to generate the constraint sensitivity information:

$$\frac{\partial \boldsymbol{g}_j}{\partial \boldsymbol{v}_i} = \boldsymbol{w}_{f_j}^T \frac{\partial \boldsymbol{R}_f}{\partial v_i} \tag{6-51}$$

where the $\frac{\partial \boldsymbol{R}_f}{\partial v_i}$ vector has been previously derived following Equation 6-42 and, again, the absence of inertia relief means that the mass terms used to generate the pseudo-load vectors are also absent.

Equation 6-51 can be expressed in matrix form to give

$$\boldsymbol{A}^T = \boldsymbol{w}_f^T \frac{\partial \boldsymbol{R}}{\partial v} \tag{6-52}$$

where $\boldsymbol{w}_f^T$ and $\frac{\partial \boldsymbol{R}}{\partial v}$ are matrices made up of vectors given in the corresponding term in Equation 6-51. The comments regarding matrix compatibility and manipulation given after Equation 6-47 apply to the Equation 6-52 calculation as well.

7. MODAL ANALYSIS

The modal analysis feature in ASTROS provides the capability to analyze and design linear structures for their modal characteristics; i.e., eigenvalues and eigenvectors. The design aspect of ASTROS places limits on the frequencies of the structures (see Subsection 2.3). The modal analysis is not only useful in its own right, but also provides the basis for a number of further dynamic analyses. Flutter and blast response analyses in ASTROS are always performed in modal coordinates. As detailed in Section 11, transient and frequency response analyses can be performed in either modal or physical coordinates, at the selection of the user.

Modal analyses typically are performed with degrees of freedom much fewer in number than static analyses. The following subsection discusses an alternative reduction procedure to the Guyan reduction technique described in Section 6. The user has the option of selecting which ASTROS option is to be used for a particular analysis. The Givens and FEER methods of eigenanalysis are also briefly discussed in this section, as are the design aspects of modal response in terms of constraint evaluation and sensitivity analysis.

7.1. GENERALIZED DYNAMIC REDUCTION

Generalized Dynamic Reduction (GDR) is a method that has been formulated for reducing degrees of freedom (DOF) by using so-called generalized DOF to represent the dynamic behavior of the structural model. The displacements of these generalized DOF are internally computed. GDR requires fewer dynamic DOF than the Guyan reduction method for comparable accuracy and, more importantly, it eliminates the burden of user selection of appropriate dynamic DOF.

GDR performs dynamic reduction by a combination of three methods: the ***Guyan reduction***, the ***inertia relief shapes*** and ***subspace iteration*** techniques. The user has the option to select any combination of the three methods. The Guyan reduction has already been discussed and can be characterized as using the static displacement shapes as the generalized DOF. When used in GDR, it allows the user to retain some of the physical DOF along with the generalized DOF. The inertia relief shapes use the displacement shapes due to the inertia loads as the generalized DOF. Finally, general subspace iteration techniques are used to compute a set of approximate eigenvectors and these approximate eigenvectors are used as the generalized DOF.

Of the three methods, the general subspace iteration technique results in the most accurate eigenvalues and eigenvectors. However, Guyan reduction is available to allow the user to retain specific physical DOF. For transient response analyses, the inertia relief shapes can be used to reduce modal truncation errors which, therefore, result in improved element stress calculations.

Physical DOF in the *f-set* are related to the generalized DOF by the following equation:

$$\boldsymbol{u}_f = \begin{Bmatrix} \boldsymbol{u}_a \\ \boldsymbol{u}_o \end{Bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{G}_{oa} & \boldsymbol{G}_{ok} & \boldsymbol{G}_{oj} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_a \\ \boldsymbol{u}_k \\ \boldsymbol{u}_j \end{Bmatrix} = \boldsymbol{G}_{fq}\ \boldsymbol{u}_q \tag{7-1}$$

where the *a-set* and *o-set* have been defined previously and

$\boldsymbol{u}_k$ are generalized DOF representing approximate eigenvectors

$\boldsymbol{u}_j$ are generalized DOF representing inertia relief shapes

$\boldsymbol{u}_q$ is the union of $\boldsymbol{u}_a$, $\boldsymbol{u}_k$, and $\boldsymbol{u}_j$

$\boldsymbol{G}_{oa}$ is the Guyan reduction constraint relationship

$\boldsymbol{G}_{ok}$ is the transformation to define the approximate eigenvectors

$\boldsymbol{G}_{oj}$ is the transformation to define the inertia relief generalized DOF

$\boldsymbol{G}_{fq}$ is the overall transformation matrix such that the stiffness matrix and the mass matrix in the generalized coordinates are

$$\boldsymbol{K}_{qq} = \boldsymbol{G}_{fq}^T \boldsymbol{K}_{ff} \boldsymbol{G}_{fq} \tag{7-2}$$

$$\boldsymbol{M}_{qq} = \boldsymbol{G}_{fq}^T \boldsymbol{M}_{ff} \boldsymbol{G}_{fq} \tag{7-3}$$

The $\boldsymbol{G}_{oa}$ matrix is identical to the $\boldsymbol{G}_o$ matrix of Equation 6-11 and therefore does not require further discussion. The $\boldsymbol{G}_{ok}$ and $\boldsymbol{G}_{oj}$ matrices are discussed in the following subsections.

7.1.1. Inertia Relief Shapes

In most transient response problems, Guyan reduction gives a reasonable approximation to the acceleration responses if the retained DOF are appropriately selected. However, the stress responses are likely to be inaccurate unless a large number of DOF are retained for analysis. One method to improve the stress responses is by using the inertia relief shapes.

The inertia relief shapes are the displacement shapes of the eliminated DOF, $\boldsymbol{u}_o$, obtained by imposing an acceleration field on the structural model. Two types are treated here: (1) inertia relief shapes due to the acceleration of the origin of the basic coordinate system and (2) inertia shapes due to an acceleration field caused by specified DOF. The user can select either or both types. The DOF to be eliminated are related to the inertia relief DOF, $\boldsymbol{u}_j$, by

$$\boldsymbol{u}_o = \boldsymbol{G}_{oj}\,\boldsymbol{u}_j = \boldsymbol{G}_{oj}^c \boldsymbol{G}_{oj}^s \begin{Bmatrix} \boldsymbol{u}_j^c \\ \boldsymbol{u}_j^s \end{Bmatrix} \tag{7-4}$$

where $\boldsymbol{u}_j^c$ denotes the inertia relief shape DOF due to coordinate acceleration and $\boldsymbol{u}_j^s$ denotes those due to acceleration caused by user specified DOF. $\boldsymbol{G}_{oj}^c$ and $\boldsymbol{G}_{of}^s$ are corresponding transformations.

The calculation of the G_{oj}^{c} matrix begins by assuming that the origin of the basic coordinate system (see Subsection 2.1) is subjected to an acceleration $\ddot{u}_c$, where $\ddot{u}_c$ has six DOF. The inertia force on the structural model is

$$F_g = M_{gg} G_{gc} \ddot{u}_c \tag{7-5}$$

where G_{gc} is a rigid body transformation matrix to transform displacements at the origin to displacements at the physical DOF and can be easily computed based on geometric data. Equation 7-5 represents an applied load where response can be computed using the same techniques as those given in Subsection 6.1. The basic equilibrium equation is

$$K_{gg} u_g = M_{gg} G_{gc} \ddot{u}_c \tag{7-6}$$

A reduction of this equation produces an equation for the u_o vector of Equation 7-4:

$$K_{oo} u_o = \left[M_{og} + T_{mo}^{T} M_{mg} \right] G_{gc} \ddot{u}_c \tag{7-7}$$

where M_{og}, T_{mo} and M_{mg} are partitions of the M_{gg} and T_{mn} (the multi-point constraint matrix) matrices. Note that the $G_o u_a$ term of Equation 6-12 is absent in this equation. This is because this term is redundant with the effects produced by the G_{oa} matrix of Equation 7-1.

The u_c vector contains the generalized DOF due to accelerations of the origin of the basic coordinate system, i.e:

$$u_j^c = u_c \tag{7-8}$$

Equation 7-7 therefore, provides the required transformation:

$$u_o = G_{oj}^{c} u_j^c \tag{7-9}$$

where

$$G_{oj}^{c} = K_{oo}^{-1} \left[M_{og} + T_{mo}^{T} M_{mg} \right] G_{gc} \tag{7-10}$$

The calculation of the G_{oj}^{s} matrix follows a similar path and starts by specifying that certain retained degrees of freedom are given a unit acceleration:

$$u_j^s = \ddot{u}_a \tag{7-11}$$

The response of the omitted degrees of freedom to acceleration is obtained from

$$K_{oo} u_o = -M_{oo} \ddot{u}_o - M_{oa} \ddot{u}_a \tag{7-12}$$

where again the effect of the displacement of the analysis set has been neglected since this information is redundant with the G_{oa} matrix.

By making the usual assumption of Guyan reduction that (cf. Equation 6-25)

$$\ddot{u}_o = G_{oa}\,\ddot{u}_a \tag{7-13}$$

then Equations 7-1, 7-11, 7-12, and 7-13 combine to give

$$G_{oj}^{s} = -K_{oo}^{-1}\left[M_{oo} + G_{oa}\,M_{oa}\right] \tag{7-14}$$

As a final note on the inertial relief shapes, experience has shown that it is necessary that the degrees of freedom which are used to create these shapes must be included as *a-set* degrees of freedom.

7.1.2. Approximate Eigenvectors

The G_{ok} matrix of Equation 7-1 contains column vectors that approximate the lowest eigenvectors of the structural modal. A general theoretical derivation of this matrix is now given and this is followed by a discussion of some of the detailed considerations that go into making this powerful technique a practical one. The discussion given here follows one given in Subsection 2.4 of the MSC/NASTRAN Application Manual of Reference 19. The reasons for this duplication are that (1) Reference 19 is relatively inaccessible and (2) there are subtle differences in the ASTROS implementation of the technique.

The standard structural eigenvalue problem is written as

$$\left[K - \lambda M\right]\varphi = 0 \tag{7-15}$$

Successive iterations of an inverse power approach for the computation of eigenvalues and eigenvectors of Equation 7-15 provide approximate eigenvectors. This approach applies a recursion relation of the form

$$\left[K - \lambda_s M\right]\,u_{i+1} = \frac{1}{c_i} M\,u_i \tag{7-16}$$

where c_i is the maximum component of u_i and λ_s is a shift point that is defined subsequently. The subspace made up of these vectors is

$$G = u_0, u_1, u_2, \ldots, u_{m-1} \tag{7-17}$$

If the complete set of eigenvectors (or modes) is given by Φ, then each of the u_n vectors can be expressed as

$$\begin{aligned} u_{n+1} &= \Phi\,\alpha_{n+1} \\ u_n &= \Phi\,\alpha_n \end{aligned} \tag{7-18}$$

The mode shapes are orthogonal so that

$$\varphi_i \boldsymbol{K} \varphi_j = \begin{cases} \lambda_i & \text{if } i=j \\ 0.0 & \text{if } i \neq j \end{cases} \tag{7-19}$$

$$\varphi_i \boldsymbol{M} \varphi_j = \begin{cases} 1.0 & \text{if } i=j \\ 0.0 & \text{if } i \neq j \end{cases} \tag{7-20}$$

If Equation 7-18 is placed into Equation 7-16 and the resulting equations are pre-multiplied by Φ^T, the Equations 7-19 and 7-20 relations give

$$\frac{\alpha_{j,n+1}}{\alpha_{j,n}} = \frac{1}{c_n\left(\lambda_j - \lambda_s\right)} \tag{7-21}$$

where the notation $\alpha_{j,n}$ indicates the j^{th} element of the α_n vector. Equation 7-21 indicates that the relative proportion of an eigenvector in successive trial vectors increases inversely to the magnitude of its shifted eigenvalue. The series therefore converges to the eigenvector closest to the shift point. The series of vectors given by Equation 7-17 are used to generate $\boldsymbol{G}_{ok}$ by setting the first column of $\boldsymbol{G}_{ok}$ to the last vector computed in the iteration process. The next to last vector is mass orthogonalized with respect to the last vector to give the second column of $\boldsymbol{G}_{ok}$. This process is repeated for preceding vectors of Equation 7-17 until the desired number of approximate eigenvectors are obtained.

Details that are needed to complete the algorithm are (1) specification of number of iterates (m in Equation 7-17), (2) specification of λ_s, (3) specification of $\boldsymbol{u}_0$, and (4) rejection of parallel vectors. Each of these is now briefly discussed.

Number of Iterates

Though the set of vectors given by Equation 7-17 should contain all the approximate eigenvectors, they are not necessarily a good basis for $\boldsymbol{G}_{ok}$. This is because some of the vectors may be parallel to one another to within the accuracy of the computer and others may be a linear combination of two or more other vectors. Therefore, it is necessary to determine more vectors in Equation 7-17 than there are eigenvectors and use the mass orthonormalization to select out an appropriate reduced set.

If λ_{max} is the highest frequency of interest, then Sturm sequence properties can be used to determine N_{max}, the number of eigenvalues below max. A safety factor of k_f is then applied to give

$$m = k_f N_{max} \tag{7-22}$$

A safety factor of 1.5 is used in ASTROS.

Determination of λ_s

Computer accuracy considerations also determine λ_s. If the range of eigenvalues varies from 0 to λ_{max}, Equation 7-21 indicates that the α values at the m^{th} iteration range from

$$\frac{\alpha_{j,m}}{\alpha_{j,o}} = \left(\frac{-\lambda_s}{\lambda_{max} - \lambda_s} \right)^m = \varepsilon \tag{7-23}$$

If the precision of the computer is less than ε, then the components of the vector series differ from one another in an insignificant, random fashion. Therefore, to ensure meaningful results, the shift value can be computed from

$$\lambda_s = \frac{\lambda_{max}\ \varepsilon_c}{1 - \varepsilon_c^{\frac{1}{m}}} \tag{7-24}$$

where 10^{-8} is selected in ASTROS as a representative value for ε_c, the precision of the computer.

Specification of Starting Vector

The $\boldsymbol{u}_0$ vector in Equation 7-17 needs to be selected so that it contains all the approximate eigenvectors. This is done by generating an initial vector using a random number generator. To provide added assurance, six distinct initial vectors are generated in this way and the orthonormalization process interweaves results from each of the six series of vectors.

Rejection of Parallel Vectors

Despite the precautions taken to ensure orthogonal vectors, it is still possible for the iterative algorithm to produce parallel results. This is checked in ASTROS by rejecting vectors whose norm is less than a specified threshold. In ASTROS, this threshold is computed by reference to Equation 7-23 and by assuming that the k_f factor will produce a maximum eigenfrequency of $k_f \lambda_{max}$. This gives a rejection threshold of

$$\varepsilon_r = \left(\frac{-\lambda_s}{-\lambda_s + k_f \lambda_{max}} \right)^n \tag{7-25}$$

and when λ_s is substituted from Equation 7-24, this gives

$$\varepsilon_r = \frac{\varepsilon_c}{\left(k_f + (1 - k_f)\, \varepsilon_c^{1/n} \right)^n} \tag{7-26}$$

Substantial testing of the dynamic reduction algorithm on large problems has shown that this value of ε_r performs well, while use of ε_c directly rejects too many candidate vectors.

7.2. THE EIGENANALYSIS METHODS

The eigenanalysis in ASTROS solves the general problem:

$$\left[\boldsymbol{K}_{aa} - \lambda \, \boldsymbol{M}_{aa} \Phi_a \right] = \mathbf{0} \tag{7-27}$$

where the a subscript is used to indicate matrices that have been obtained by the Guyan reduction of Equations 6-14 or from the Dynamic Reduction to the *q-set* of Equations 7-2 and 7-3. Three methods of eigenanalysis are available: the Inverse Power method; the Given's method; and the FEER method. Each of these are described in the following subsections.

7.2.1. The Inverse Power Method

The Inverse Power method with shifts (Reference 1) is an iterative procedure applied directly to 7-27. It is required to find all of the eigenvalues and eigenvectors with a specified range of λ. Let:

$$\lambda = \lambda_o + \Lambda \tag{7-28}$$

where λ_o is a constant called the shift point. Therefore, Λ replaces λ as the eigenvalue of the system. The algorithm for the *nth* iteration step is defined by:

$$\left[\boldsymbol{K}_{aa} - \lambda_o \, \boldsymbol{M}_{aa} \right] \boldsymbol{w}_n = \boldsymbol{M}_{aa} \frac{1}{c_{n-1}} \boldsymbol{w}_{n-1} \tag{7-29}$$

where c_n , a scalar, is equal to the element of the vector $\boldsymbol{w}_n$ with the largest absolute value. At convergence, $1/c_n$ converges to Λ , the shifted eigenvalue closest to the shift point and $\boldsymbol{w}_n$ converges to the corresponding eigenvector φ . Note from 7-29 that a triangular decomposition of $\left[\boldsymbol{K}_{aa} - \lambda_o \, \boldsymbol{M}_{aa} \right]$ is necessary in order to evaluate $\boldsymbol{w}_n$. The shift point, λ_o , can be changed in order to improve the rate of convergence toward a particular eigenvalue, or to improve accuracy and convergence rates after several roots have been extracted fromabout a given shift point. Also, λ_o , can be calculated such that the eigenvalues within a desired frequency band can be found rather than those that have the smallest absolute value.

For calculating additional eigenvalues, the trial vectors $\boldsymbol{w}_n$ in 7-29 must be swept to eliminate contributions due to previously found eigenvalues that are closer to the shift point than the current eigenvalue. The algorithm to do this is:

$$\varphi_n = \overline{\boldsymbol{w}}_n - \sum_{i=1}^{m} \left[\overline{\varphi}_i^{\,t} \, \boldsymbol{M} \, \overline{\boldsymbol{w}}_n \right] \overline{\varphi}_i \tag{7-30}$$

where $\overline{w}_n$ is the trial vector being swept, m is the number of previously swept eigenvalues, and $\overline{\varphi}_i$ is defined by:

$$\overline{\varphi}_i = \frac{w_{i,N}}{\sqrt{w_{i,N}^t M w_{i,N}}} \tag{7-31}$$

where $w_{i,N}$ is the last eigenvector found in iterating for the *ith* eigenvalue.

7.2.2. The Given's Method

The Givens, or Tridiagonal, method of eigenanalysis is also available. This well known algorithm is briefly summarized here, with more detailed information available in Subsections 9.2 and 10.2 of Reference 1 and Subsection 13.5 of Reference 14.

If there are rigid body modes, it is recommended that the support concepts of Equations 6-16 through 6-18 be used to define these modes. The calculated rigid body modes are:

$$\Phi_r^T m_r \Phi_r = I \tag{7-32}$$

and m_r is the rigid body mass matrix of Equation 6-21.

The Given's method of eigenanalysis proceeds into six steps outlined below:

1. The mass matrix is decomposed into Choleski factors:

$$M_{aa} = C C^T \tag{7-33}$$

and this is substituted into Equation 7-27 to give:

$$K_{aa} = \lambda C C^T \Phi_a = 0 \tag{7-34}$$

2. Intermediate vectors are defined as

$$a = C^T \Phi_a \tag{7-35}$$

and Equation 7-34 is multiplied by C^{-1} to give

$$J - \lambda I \; a = 0 \tag{7-36}$$

where

$$J = C^{-1} K_{aa} C^{-T} \tag{7-37}$$

and the $-T$ indicates inverse transpose.

3. The ***J*** matrix is then reduced to tridiagonal form using the Given's method as described in Subsection 10.2.2 of Reference 1.

4. A ***Q-R*** iterative algorithm is then used to further transform this matrix to a diagonal form, where the diagonal terms are the eigenvalues of the system.

5. The extracted eigenvalue is then substituted in Equation 7-36 and solved for the corresponding eigenvector. The number of eigenvectors that are to be determined is specified by the user. Although it would appear that this equation could be solved by direct substitution, this technique has been shown to be unpredictable and an alternative, iterative procedure based on an algorithm given on pages 315 - 330 of Reference 20 is used.

6. The eigenvector in the *a-set* degrees of freedom are calculated based on Equation 7-35:

$$\Phi_a = \boldsymbol{C}^{-1}\boldsymbol{a} \tag{7-38}$$

Recovery of the modes to the global set is similar to that given for the displacement recovery in Subsection 6.1. If Dynamic Reduction has been used, the *f-set* degrees of freedom are calculated from Equation 7-1 while a similar recovery is used for Guyan Reduction:

$$\Phi_f = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{G}_o \end{bmatrix} \boldsymbol{a}\ \Phi_a \tag{7-39}$$

where G_o is given by Equation 6-11.

Recovery to the *n-set* entails merging in any enforced displacements while the *m-set* displacements are obtained in a manner similar to Equation 6-2

$$\Phi_m = \boldsymbol{T}_{mn}\,\Phi_n \tag{7-40}$$

The eigenvectors in the *g-set* are then obtained by merging *m-set* and *n-set* DOF

$$\Phi_g = \begin{bmatrix} \Phi_m \\ \Phi_n \end{bmatrix} \tag{7-41}$$

7.2.3. The FEER Method

The FEER Method (or Tridiagonal Reduction) is a matrix reduction scheme wherein the eigenvalues in the neighborhood of a specified point, λ_o, in the eigenspectrum can be accurately determined from a tridiagonal eigenvalue problem whose dimension, or order, is much smaller than the full problem. The order of the reduced problem, m, obeys the relation $m \leq 2\,\overline{q} + 10$ where $\overline{q}$ is the desired number of eigenvalues. Thus, the power of the FEER method lies in the fact that the size of the reduced problem is the same order as the number of desired roots, even though the finite element model may have thousands of degrees of freedom.

The five basic steps in the FEER method are summarized below:

1. Convert the generalized eigenproblem:

$$\boldsymbol{K}_{aa}\,\varphi_a = \lambda \boldsymbol{M}_{aa}\,\varphi_a \tag{7-42}$$

to the symmetric inverse form:

$$\boldsymbol{B}\,\varphi_a = \left[\frac{1}{\lambda - \lambda_o}\right]\boldsymbol{M}_{aa}\,\varphi_a \tag{7-43}$$

B is derived by recognizing that when vibration modes are requested in the neighborhood of a specified frequency, λ_o, Equation 7-42 may be rewitten as:

$$\left[\boldsymbol{K}_{aa} - \lambda_o \boldsymbol{M}_{aa}\right]\varphi_a = \left(\lambda - \lambda_o\right)\boldsymbol{M}_{aa}\,\varphi_a \tag{7-44}$$

Now, let $\overline{\boldsymbol{K}} = \boldsymbol{K} - \lambda_o \boldsymbol{M}$ and $\lambda' = \lambda - \lambda_o$. Then, Equation 7-44 may be rearranged to yield:

$$\boldsymbol{M}\overline{\boldsymbol{K}}^{-1}\boldsymbol{M}\,\varphi = \frac{1}{\lambda'}\boldsymbol{M}\,\varphi \tag{7-45}$$

A Cholesky decomposition is then used to factor $\overline{\boldsymbol{K}}$:

$$\overline{\boldsymbol{K}} = \boldsymbol{L}\,\boldsymbol{d}'\,\boldsymbol{L}^T \tag{7-46}$$

where $\boldsymbol{L}$ is the lower triangular matrix and $\boldsymbol{d}'$ is a diagonal matrix. Then Equation 7-45 may be written:

$$\boldsymbol{M}\left[\,\boldsymbol{L}^{-T}\boldsymbol{d}'^{-1}\boldsymbol{L}^{-1}\right]\boldsymbol{M}\,\varphi = \frac{1}{\lambda'}\,\boldsymbol{M}\,\varphi \tag{7-47}$$

or, simplifying:

$$\boldsymbol{B}\,\varphi = \overline{\Lambda}\boldsymbol{M}\,\varphi \tag{7-48}$$

where

$$\boldsymbol{B} = \boldsymbol{M}\left[\,\boldsymbol{L}^{-T}\boldsymbol{d}'^{-1}\boldsymbol{L}^{-1}\right]\boldsymbol{M}$$

$$\Lambda = \frac{1}{\lambda'} = \frac{1}{\lambda - \lambda_o}.$$

2. The tridiagonal reduction is use to transform 7-43 into a tridiagonal form of reduced order. To do this, Equation 7-43 is rewritten as:

$$\overline{\boldsymbol{B}}\,\varphi = \Lambda\,\varphi \tag{7-49}$$

where $\overline{\boldsymbol{B}} = \boldsymbol{M}^{-1}\boldsymbol{B}$. Now, $\overline{\boldsymbol{B}}$ is reduced to tridiagonal form, $\boldsymbol{A}$, using a single vector Lanczos recurrance defined by:

$$\left.\begin{aligned} a_{ii} &= \boldsymbol{V}_i^T \boldsymbol{B}\, \boldsymbol{V}_i \\ \overline{\boldsymbol{V}}_{i+1} &= \overline{\boldsymbol{V}}\, \boldsymbol{V}_i - a_{ii}\boldsymbol{V}_i - d_i \boldsymbol{V}_{i-1} \\ d_{i+1} &= \left[\, \overline{\boldsymbol{V}}_{i+1}^T \boldsymbol{M}\, \overline{\boldsymbol{V}}_{i+1} \right]^{1/2} \end{aligned}\right\} \quad i = 1, 2, \ldots, m \tag{7-50}$$

$$\boldsymbol{V}_{i+1} = \frac{1}{d_{i+1}} \overline{\boldsymbol{V}}_{i+1} \qquad i = 1, 2, \ldots, m-1$$

where vector $\boldsymbol{V}_o$ is set to zero, $\boldsymbol{V}_1$ is a random vector, and $d_1 = 0$. The reduced tridiagonal problem resulting is then:

$$\boldsymbol{A}\boldsymbol{y} = \begin{bmatrix} a_{11} & d_2 & & & & & \\ d_2 & a_{22} & d_3 & & & & \\ & d_3 & a_{33} & d_4 & & & \\ & & \cdots & \cdots & \cdots & & \\ & & & & d_{m-1} a_{m-1,m-1} & d_m & \\ & & & & d_m & a_{mm} & \end{bmatrix} \boldsymbol{y} = \overline{\Lambda}\boldsymbol{y} \tag{7-51}$$

where $\overline{\Lambda}$ approximates the eigenvalue Λ of Equation (7-43), and $\boldsymbol{y}$ is an eigenvector of $\boldsymbol{A}$. These recurrance relations generate a matrix, V, vector by vector, i.e.

$$\boldsymbol{V} = \left[\, \boldsymbol{V}_1, \boldsymbol{V}_2, \ldots, \boldsymbol{V}_m \right] \tag{7-52}$$

and Equations (7-50) are modified such that each vector $\boldsymbol{V}_{i+1}$ is re-orthogonalized to all previously computed vectors, that is, V is orthonormal to M:

$$\boldsymbol{V}^T \boldsymbol{M}\, \boldsymbol{V} = \boldsymbol{I} \tag{7-53}$$

Thus,

$$\boldsymbol{A} = \boldsymbol{V}^T \boldsymbol{B}\, \boldsymbol{V} \tag{7-54}$$

which results in $\boldsymbol{A}$ having order m.

3. The eigenvalues, $\overline{\Lambda}$, and eigenvectors, $\boldsymbol{y}$, of the reduced matrix are extracted using a Q-R algorithm as indicated in the previous section. The eigenvectors are normalized so that:

$$\boldsymbol{y}_i^T \boldsymbol{y}_i = 1 \qquad i = 1, \ldots, m \tag{7-55}$$

4. Upper and lower bounds on the eigenvalues are obtained. This is done using the following error bound formula which serves as a criterion for selecting acceptable eigensolutions:

$$\varepsilon_i = 1 - \frac{\bar{\lambda}_i}{\lambda_i} \leq \frac{d_{m+1} y_{mi}}{\bar{\Lambda}_i \left(1 + \lambda_o \bar{\Lambda}_i \right)} \tag{7-56}$$

5. The corresponding eigenvectors are computed and converted to the physical set. If $(\bar{\Lambda}, y)$ is an eigenpair of Equation 7-43, then:

$$\boldsymbol{A}\boldsymbol{y} = \bar{\Lambda}\boldsymbol{y} \tag{7-57}$$

or, from Equations 7-53 and 7-54:

$$\begin{aligned} \boldsymbol{V}^T\boldsymbol{B}\boldsymbol{V}\boldsymbol{y} &= \bar{\Lambda}\boldsymbol{V}^T\boldsymbol{M}\boldsymbol{V}\boldsymbol{y} \\ \boldsymbol{B}\boldsymbol{V}\boldsymbol{y} &= \bar{\Lambda}\boldsymbol{M}\boldsymbol{V}\boldsymbol{y} \end{aligned} \tag{7-58}$$

If $\varphi = \boldsymbol{V}\boldsymbol{y}$, then $\boldsymbol{B}\varphi = \bar{\Lambda}\boldsymbol{M}\varphi$, hence $(\bar{\Lambda}, \varphi)$ is an eigenpair of 7-43 and the corresponding eigenvector may be calculated from:

$$\varphi = \boldsymbol{V}\boldsymbol{y} \tag{7-59}$$

and the eigenvalue λ is calculated from Equation 7-43 using:

$$\lambda = \frac{1}{\Lambda} + \lambda_o \tag{7-60}$$

Note that the matrix $\boldsymbol{B}$ enters the recurrance formulae, Equations 7-50, only through the matrix-vector multiplications $\boldsymbol{B}\boldsymbol{V}_i$. Therefore, $\boldsymbol{B}$ is not modified by the computations.

7.3. CONSTRAINT EVALUATION

Given the eigenvalues, the constraint values are determined as:

$$g_j = 1.0 - \frac{(2\pi f_{\text{high}})^2}{\lambda_j} \tag{7-61}$$

for upper bound constraints and

$$g_j = \frac{(2\pi f_{\text{high}})^2}{\lambda_j} - 1.0 \tag{7-62}$$

for lower bound constraints, where f_{high} and f_{low} are the frequency limits as specified in Equation 4-24 and λ_j is the extracted eigenvalue. The extracted value has been placed in the denominator because there

is a desire (see Subsection 13.1) to express constraints in a form that make them linear in the inverse of the design variable. The assumption made here is that non-structural mass makes the eigenvalue much more sensitive to changes in the structural stiffness than to mass changes. The stiffness, in turn, is assumed to be a linear function of the design variable. Obviously, there are cases where these assumptions do not apply.

7.4. FREQUENCY CONSTRAINT SENSITIVITIES

The calculation of sensitivities of frequency constraints to changes in design variables begins by differentiating Equation 7-61 or 7-62. For Equation 7-61, this gives

$$\frac{\partial g_j}{\partial v_i} = \frac{(2\pi f_{\text{high}})^2}{\lambda_j^2} \frac{\partial \lambda_j}{\partial v_i} = \frac{1.0 - g_j}{\lambda_j} \frac{\partial \lambda_j}{\partial v_i} \tag{7-63}$$

The determination of $\frac{\partial \lambda_j}{\partial v_i}$ is performed using well known relationships (Reference 21) that can be represented conceptually by starting with the basic modal equation:

$$\left[\boldsymbol{K} - \lambda_j \boldsymbol{M} \right] \varphi_j = \boldsymbol{0} \tag{7-64}$$

Taking the derivative of 7-64 with respect to v_i gives

$$\left[\frac{\partial \boldsymbol{K}}{\partial v_i} - \frac{\partial \lambda_j}{\partial v_i} \boldsymbol{M} - \lambda_j \frac{\partial \boldsymbol{M}}{\partial v_i} \right] \varphi_i + \left[\boldsymbol{K} - \lambda \boldsymbol{M} \right] \frac{\partial \varphi_j}{\partial v_i} = \boldsymbol{0} \tag{7-65}$$

If this equation is premultiplied by φ_j^T and the self-adjoint nature of the symmetric eigenvalue problem is utilized, i.e.,

$$\varphi_j^T \left[\boldsymbol{K} - \lambda_j \boldsymbol{M} \right] = \boldsymbol{0} \tag{7-66}$$

then Equation 7-65 becomes

$$\frac{\partial \lambda_j}{\partial v_i} = \frac{\varphi_j^T \left[\frac{\partial \boldsymbol{K}}{\partial v_i} - \lambda_j \frac{\partial \boldsymbol{M}}{\partial v_i} \right] \varphi}{\varphi_j^T \boldsymbol{M} \varphi_j} \tag{7-67}$$

Equation 7-67 is evaluated in ASTROS in the *g-set*, thereby allowing use of the $\frac{\partial \boldsymbol{K}_{gg}}{\partial v}$ and $\frac{\partial \boldsymbol{M}_{gg}}{\partial v}$ matrices of Equations 5-41 and 5-42 and the vectors of φ_g of Equation 7-41 that are associated with the constrained eigenvalues.

THIS PAGE INTENTIONALLY LEFT BLANK

8. AERODYNAMIC ANALYSES

Accurate aerodynamic analyses are a critical component in the performance of the multidisciplinary analysis capability contained in ASTROS. This section describes the generation of the steady and unsteady aerodynamic matrices that are present in ASTROS while subsequent sections describe the application of these aerodynamics. The splining techniques that are used to couple the aerodynamic and structural models are also described in this section.

8.1. STEADY AERODYNAMICS

Steady aerodynamics are used in ASTROS for the computation of loads on an aircraft structure. The selection of an appropriate algorithm for computing these forces is not an easy task since methods vary in complexity from "back-of-the-envelope" calculations to sophisticated computational fluid dynamics algorithms. The USSAERO (Unified Subsonic and Supersonic Aerodynamic Analysis) algorithm of Reference 22 was selected primarily because it represents an algorithm of medium complexity, consistent with the preliminary design role of ASTROS, and because it is an algorithm that has been used extensively in the performance of aerodynamic and aeroelastic analysis. In particular, the USSAERO code had been integrated with a dynamic structural response capability in the performance of an Air Force supported contract in the area of maneuver loads (Reference 23) and this experience was directly applicable to the ASTROS integration task.

8.1.1. USSAERO Capabilities

USSAERO determines the pressure distributions on lifting wing-body-tail combinations using numerical methods. The solid boundaries are represented by a number of discrete panels as depicted in Figure 19. The flow around the solid boundaries can be estimated by the superposition of source type singularities for non-lifting bodies and vortex singularities for wing-like singularities. The USSAERO algorithm has undergone a number of updates and only a subset of the total capabilities have been implemented in ASTROS. Therefore, it is necessary that the capabilities of the ASTROS implementation be defined. Among the features supported are:

1. Subsonic and supersonic analyses.
2. Symmetric and antisymmetric analyses.
3. Multiple lifting surfaces, both coplanar and non-coplanar.
4. Body elements can be used to represent fuselage and pod (e.g., nacelles or stores) components.
5. Pitch, roll and yaw control surfaces can be specified.
6. Pitch, roll and yaw rates can be specified.
7. Thickness and camber effects on the lifting surfaces.

8. Aerodynamic influence coefficients on both wing and body components (AIC).

It is equally useful to list capabilities that have been installed in USSAERO versions that are not supported in ASTROS:

1. There is no asymmetric capability, either in terms of the configuration or the aerodynamic forces.
2. The nonplanar option for representing thick lifting surfaces has only been supported for rigid aerodynamics analysis. This option is inconsistent with the aerodynamic influence coefficient requirements for ASTROS aeroelastic analysis.

Subsection 3.3 of the Applications Manual contains guidelines for generating aerodynamic models and therefore, has more specific information about the USSAERO capabilities in ASTROS.

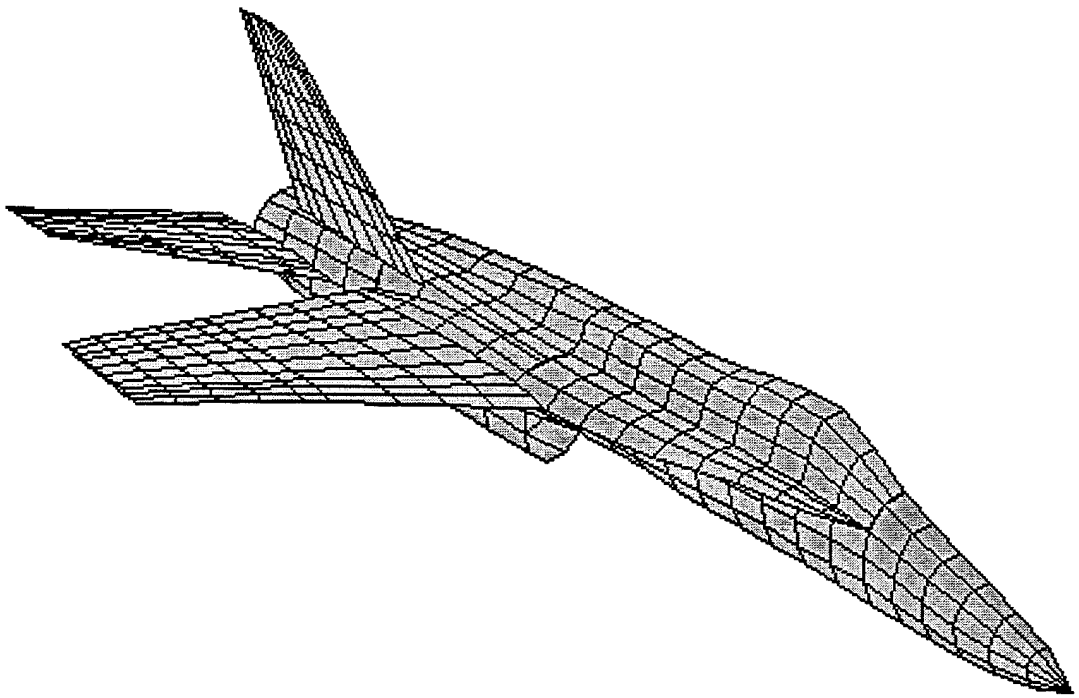


Figure 19. Aerodynamic Paneling in USSAERO

8.1.2. USSAERO Methodology

The formulation of the methodology used in ASTROS is contained in Reference 22 while this writeup provides an overview which defines the aerodynamic matrices which are generated for the steady aeroelastic analyses.

The basic equation in USSAERO is given by:

$$\begin{bmatrix} A_{bb} & A_{bw} \\ A_{wb} & A_{ww} \end{bmatrix} \begin{Bmatrix} \sigma \\ \gamma \end{Bmatrix} = \begin{Bmatrix} \omega_b \\ \omega_w \end{Bmatrix} \tag{8-1}$$

where

b	denotes the body
w	denotes the lifting surface
ω	are velocities at the panels due to a prescribed boundary condition
σ	are source singularities on the body
γ	are vortex singularities on the lifting surfaces
$\boldsymbol{A}$	are normal velocity influence coefficients

Terms in the $\boldsymbol{A}$ matrix provide the normal velocity that is produced at a receiving panel due to a unit value of the singularity at a sending panel. This matrix can be computed from the superposition of individual velocity influence coefficients, which in turn can be computed from geometric considerations and the prescribed Mach number. The boundary conditions can account for airfoil camber and thickness, angle of attack, control surface settings and aircraft rates.

Once the values of the singularities have been determined, the velocity components can be computed and pressure coefficients at each of the panels are calculated using:

$$C_{P_i} = \frac{-2}{\gamma M^2} \left\{ \left[1 + \frac{\gamma - 1}{2} M^2 \left(1 - q_i^2\right) \right]^{\frac{\gamma}{\gamma - 1}} - 1 \right\} \tag{8-2}$$

where

M	is the Mach number
γ	is the Specific heat ratio
q_i^2	$= u_i^2 + v_i^2 + w_i^2$
u_i	$= u_o + \Delta u_i$ i.e. the backwash
v_i	$= v_o + \Delta v_i$ i.e. the sidewash
w_i	$= w_o + \Delta w_i$ i.e. the upwash

u_o, v_o and w_o are the components of the onset flow in the reference axis system and are normalized with respect to the freestream velocity. Perturbation velocities at each panel, Δu_i, Δv_i, Δw_i, are also normalized with respect to the freestream velocity. For lifting surfaces, the calculation of Equation 8-2 is repeated for the upper and lower surfaces.

As a final step, these pressure coefficients are dimensionalized and converted to forces. These forces are output in matrix ***AIRFRC***, the rows of which are the panels and the columns correspond to individual boundary conditions. This matrix is discussed further in Subsection 9.1.

The ***AIRFRC*** matrix provides loads that are applicable if the aircraft is structurally rigid. A second matrix, ***AIC***, is generated in the USSAERO module to provide for the incremental loads created by the structural deformations. This matrix is generated in ASTROS by making the approximation that the pressure expression of Equation 8-2 is

$$C_{P_i} = \frac{-2\,\Delta u_i}{U_\infty} \tag{8-3}$$

This equation is developed by assuming that $u_o = 1, \Delta u_i << 1, v_i << 1, w_i << 1$ and uses the mathematical approximation that $(1+\varepsilon)^a \approx 1 + a\,\varepsilon$.

The total force on a wing panel can be derived from Equation 8-3.

$$F_i = C_{P_i} A_i = \frac{-4\,\Delta u_i A_i}{U_\infty} \tag{8-4a}$$

where A_i is the area of the panel and equal contributions from the upper and lower surface account for the factor of two. Similarly, for bodies:

$$F_i = \frac{-2\,\Delta u_i A_i}{U_\infty} \tag{8-4b}$$

The ***AIC*** matrix calculation is then

$$\boldsymbol{AIC} = -2\begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A} \end{bmatrix} \boldsymbol{U} \begin{bmatrix} \boldsymbol{A}_{bb} & \boldsymbol{A}_{bw} \\ \boldsymbol{A}_{wb} & \boldsymbol{A}_{ww} \end{bmatrix}^{-1} \tag{8-5}$$

where $\boldsymbol{U}$ is the influence coefficient matrix for the velocity in the streamwise direction due to singularities of the panels:

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{A}_v & \boldsymbol{B}_v \\ 2\,\boldsymbol{C}_v & 2\,\boldsymbol{D}_v \end{bmatrix} \tag{8-6}$$

where the A_v matrix gives the velocities on body panels due to singularities on the body, the B_v matrix gives the velocities on the body panels due to singularities on the wing, and the C_v matrix gives the velocities on the wing panels due to singularities on the body and D_v gives velocities on the wing panels due to singularities on the wing.

In the context of multidisciplinary design, a single design task may require analyses at a number of Mach numbers and both symmetric and antisymmetric conditions. This is accommodated in ASTROS by creating separate ***AIC*** and ***AIRFRC*** matrices for each Mach number required in the task and, for antisymmetric analyses, creating an ***AAIC*** matrix which is generated by differencing contributions from the left and right sides of the aircraft (rather than adding them for symmetric analyses) in the A matrix of Equation 8-1 and the U matrix of Equation 8-6.

8.2. UNSTEADY AERODYNAMICS

Unsteady aerodynamics are used for a variety of purposes in ASTROS, each of which has its own requirements. The flutter analysis requires unsteady aerodynamic influence coefficients to integrate the effects of the structural deformations and the aerodynamic forces in an assessment of dynamic stability. The gust analysis requires aerodynamic forces, both to generate the loads that the gust creates on the structure and to estimate the aeroelastic effects in the response to this load. The blast analysis is similar to the gust analysis, but the methodology for the blast analysis integrated into ASTROS requires these matrices in a slightly different form (See Appendix B).

Because there are fewer candidates, the selection of the algorithms to provide the unsteady aerodynamic operators was simplified, relative to the steady case. For subsonic applications, the Doublet Lattice Method (DLM) algorithm of Reference 24 was selected because it has become an industry standard and because its implementation in NASTRAN provided a resource for ASTROS code development. For supersonic applications, a comparable standard algorithm does not exist, but an obvious candidate did emerge: the constant pressure method (CPM) of Reference 25. The primary attraction of CPM is that its geometrical input and its matrix output is consistent with doublet lattice so that the majority of the code required for the two algorithms can be shared. Another attribute is that Northrop has tested the CPM algorithm extensively, with favorable results (Reference 26). In particular, CPM's capability to address interfering and intersecting surfaces was shown to perform well. As in the steady aerodynamics case, the referenced documents are cited as sources of detailed information on methodology employed in these algorithms. This manual emphasizes the generation of matrices required in ASTROS applications.

8.2.1. Unsteady Aerodynamics Capabilities

The DLM and CPM procedures calculate matrices which provide forces on panels as a function of deflections at these panels. As this implies, the discretization of an aircraft into a number of panels, in a fashion similar to the steady aerodynamics model of Figure 15, is the basis for these methods. Capabilities of the codes include:

1. Symmetric, antisymmetric and asymmetric analyses with respect to the aircraft centerline are available.

2. Symmetric and asymmetric analysis with respect to the x-y plane is also provided by DLM. Symmetric analysis represents a ground effect option. Only asymmetric analyses are available in CPM.
3. The DLM permits the use of slender body theory and interference panels to model the effects of bodies. Bodies are not modeled in CPM.
4. Multiple lifting surfaces can be analyzed.
5. No thickness or camber effects are included in unsteady analyses so that lifting surfaces are analyzed as flat plates.

8.2.2. Unsteady Aerodynamics Methodology

The essence of the unsteady aerodynamics methods resides in the development of three basic matrices (see Subsection 17.5 of Reference 1):

$$\boldsymbol{w} = \boldsymbol{A}\boldsymbol{P} \tag{8-7}$$

$$\boldsymbol{w} = \boldsymbol{D}\boldsymbol{u} \tag{8-8}$$

$$\boldsymbol{F} = \boldsymbol{S}\boldsymbol{P} \tag{8-9}$$

where

$\boldsymbol{w}$	is the downwash (normal wash) at the aerodynamic control point
$\boldsymbol{A}$	is the aerodynamic influence matrix (ASTROS actually computes $\boldsymbol{A}^T$)
$\boldsymbol{P}$	is the pressure on the aerodynamic panel at the vortex line
$\boldsymbol{D}$	is a "Substantial differentiation" matrix
$\boldsymbol{u}$	are the displacements at the aerodynamic grid points
$\boldsymbol{F}$	are forces and moments at the aerodynamic grid points
$\boldsymbol{S}$	is an integration matrix

The goal of the unsteady aerodynamic theory is to determine the forces due to a given set of displacements. Simply stated, this is done by first determining the downwash using Equation 8-8, then solving for the pressure corresponding to this downwash using Equation 8-7 and a predetermined $\boldsymbol{A}$ matrix and, finally, using Equation 8-9 to integrate the pressures over the panels to determine the forces. The details of this development are substantially more involved and will not be presented here. In particular, the development of the $\boldsymbol{A}$ matrix involves integrations of a kernel over the lifting surfaces. The presence of bodies further complicates this evaluation. For purposes of this discussion, it suffices to say that the $\boldsymbol{A}$ matrix is a function of both Mach number and reduced frequency $k = \frac{\omega b}{U}$, where ω is the frequency of oscillation, U is the free stream velocity and b is the length of a reference semi-chord). The $\boldsymbol{D}$ matrix is a straightforward function of the panel geometry (with the exception noted in the following paragraph) with real and imaginary components corresponding to the spatial and time derivatives of the displacements. The $\boldsymbol{S}$ matrix is a simple function of geometry when only lifting surfaces are present, but becomes a function of M and k when bodies are present and has a separate representation for subsonic and supersonic Mach numbers.

The implementation of the unsteady aerodynamics method occurs in two stages: (1) Generation of geometry and related information and (2) Generation of the aerodynamic matrices $\boldsymbol{A}$, $\boldsymbol{D}$, and $\boldsymbol{S}$ of Equations 8-7 through 8-9. The definitions associated with these equations specify three points for each panel: the aerodynamic control point, the vortex line and the aerodynamic grid point. The location of each of these points, as a percentage of panel chord, is given in Table 9. This information is key to the proper generation of the $\boldsymbol{S}$ and $\boldsymbol{D}$ matrices.

The application of the $\boldsymbol{A}$, $\boldsymbol{D}$, and $\boldsymbol{S}$ matrices requires further, discipline dependent, processing. Additional relations that are required for this processing include:

$$\boldsymbol{u}_a = \boldsymbol{UG}\ \boldsymbol{u}_s \tag{8-10}$$

$$\boldsymbol{F}_s = \boldsymbol{UG}^T\ \boldsymbol{F}_a \tag{8-11}$$

$$\boldsymbol{u}_s = \Phi\ \boldsymbol{q}_s \tag{8-12}$$

$$\boldsymbol{F}_q = \Phi^T\ \boldsymbol{F}_s \tag{8-13}$$

In this idiosyncratic notation, the a subscript refers to aerodynamic degrees of freedom and s refers to structural degrees of freedom.

Table 10 identifies all the terms used in Equations 8-7 through 8-13 and gives their dimensions, where the sizes refer to:

nj	total number of aerodynamic panels
nk	total number of degrees of freedom in the aerodynamic coordinate system
na	number of degrees of freedom in the user's analysis set
nm	number of retained modes

The value of nk is typically two times nj, but bodies may add additional degrees of freedom. The spline matrix, $\boldsymbol{UG}$, is discussed in Subsection 8.3 while the normal modes are discussed in Subsection 7.2

For flutter and gust analyses, a generalized aerodynamic force matrix is computed for each Mach number and reduced frequency:

Table 9. Aerodynamic Panel Points

POINT	METHOD	
	DOUBLET LATTICE	CONSTANT PRESSURE
VORTEX	0.25	0.50
GRID	0.50	0.50
CONTROL	0.75	0.95

Table 10. Matrices For Unsteady Aerodynamic Forces

MATRIX	NO. OF ROWS	NO. OF COLUMNS	TYPE	DESCRIPTION
Φ	na	nm	Real	Retained normal modes
UG	nk	na	Real	Spline matrix relating aerodynamics to structural dof's
S	nk	nj	Real	Integration matrix
A	nj	nj	Complex	Aerodynamic influence matrix
D	nj	nk	Complex	Substantial derivative matrix
u_S	na	1	Complex	Displacements at structural points
u_a	nk	1	Complex	Displacements at aero grid points
q_s	nm	1	Complex	Modal generalized coordinates
F_s	na	1	Complex	Forces at structural points
F_a	nk	1	Complex	Forces at aerodynamic grids
F_q	nm	1	Complex	Generalized forces

$$Q_{hh} = \Phi^T \, UG^T \, S \, A^{-1} D \, UG \, \Phi \tag{8-14}$$

The design loop of ASTROS makes it efficient to break this calculation into steps that are independent of the structural design and those that are dependent. For example, the $S A^{-1} D$ matrix is independent of the structure and is therefore calculated once in the preface portion of ASTROS and is identified as Q_{kk}. The spline matrix is independent of the structure in the g-set, but goes through set reductions which depend on the stiffness and therefore the reduced spline matrix is recalculated after each design. The normal modes, of course, are a strong function of the design and are completely recalculated for each design iteration.

Gust analyses, as discussed in Subsection 11.2.3, require an additional matrix for each Mach number and reduced frequency:

$$Q_{hj} = \Phi^T \, UG^T \, S A^{-1} \tag{8-15}$$

This matrix is also computed in stages, with $S A^{-1}$ identified as Q_{kj}.

Blast analyses, as discussed in Appendix B, require A^{-1} directly. Appendix B and Section 12 discusses further processing of the aerodynamic forces.

8.3. CONNECTING AERODYNAMIC AND STRUCTURAL MODELS

The steady and unsteady aerodynamics quantities are computed at aerodynamic grids that typically do not coincide with the structural grid points. The transfer of displacements and forces from one set of grids to the other has been a troublesome task, with no universally accepted technique. ASTROS has implemented three techniques, with the primary interconnection algorithms being the surface spline and beam spline techniques of Reference 27 and Reference 1, respectively. A third algorithm performs a simple equivalent force transformation from the aero panels to a specified structural grid. Each of these algorithms is now discussed.

8.3.1. Surface Spline

The methodology associated with this spline is simple enough that its derivation, as given in Subsection 17.3.1 of Reference 1 is essentially repeated here.

A surface spline is used to find a function $w(x, y)$ for all points (x, y) when w is known for a discrete set of points, $w_i = w(x_i, y_i)$. An infinite plate is introduced to solve for the total deflection pattern given deflections at a discrete set of points. This surface spline is a smooth continuous function which is nearly linear in x and y at large distances from the points (x_i, y_i). Furthermore, the problem can be solved in closed form.

The deflection of the plate is synthesized as the response due to a set of point loads on the infinite plate. The response due to a single load is called a fundamental solution. The fundamental solutions have polar symmetry. If the load is taken at $x_i = y_i = 0$, and polar coordinates are used $x = r\cos\theta$, $y = r\sin\theta$ the governing differential equation is

$$D\ \nabla^4 w = D\frac{1}{r}\frac{d}{dr}\left\{ r\frac{d}{dr}\left[\frac{1}{r}\frac{d}{dr} r\frac{dw}{dr}\right]\right\} = q \tag{8-16}$$

The load q vanishes except near r = 0. A solution to the general spline problem, formed by super-imposing solutions of Equation 8-16 is given by

$$w(x,y) = a_0 + a_1 x + a_2 y + \sum_{i+1}^{N} K_i(x,y)\, P_i \tag{8-17}$$

where

$$K_i(x,y) = \left(\frac{1}{16\,\pi D}\right) r_i^2 \ln r_i^2, \quad r_i^2 = (x - x_i)^2 + (y - y_i)^2$$

and

$$P_i = \text{concentrated at } \left(x_i\ ,\ y_i\right)$$

The $N+3$ unknowns ($a_0, a_1, a_2, P_i,\ i=1, N$) are determined from the $N+3$ equations

$$\sum P_i = \sum x_i P_i = \sum y_i P_i = 0$$

and

$$w_j = a_0 + a_1 x_j + a_2 y_j + \sum_{i=1}^{N} K_{ij} P_i \qquad (j=1, N) \tag{8-18}$$

where

$$K_{ij} = K_i\left(x_j, y_j\right)$$

Note that $K_{ij} = K_{ji}$, and that $K_{ij} = 0$ when $i = j$. The details of the derivation are given in Reference 27.

These equations can be summarized in matrix form

$$w(x,y) = \left[\, 1, x, y, \ \boldsymbol{K}_1(x,y), \boldsymbol{K}_2(x,y), ..., \boldsymbol{K}_N(x,y) \,\right] \begin{Bmatrix} \boldsymbol{a}_0 \\ \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \hline \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \\ \cdot \\ \cdot \\ \boldsymbol{P}_N \end{Bmatrix} \tag{8-19}$$

The vector of a's and P's is found by solving

$$\begin{Bmatrix} 0 \\ 0 \\ 0 \\ \\ w_1 \\ w_2 \\ \cdot \\ \cdot \\ w_N \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & \cdot & \cdot & 1 \\ 0 & 0 & 0 & x_1 & \cdot & \cdot & x_N \\ 0 & 0 & 0 & y_1 & \cdot & \cdot & y_N \\ & & & & & & \\ 1 & x_1 & x_1 & 0 & \cdot & \cdot & K_{1N} \\ 1 & x_2 & x_2 & \cdot & \cdot & \cdot & K_{2N} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & x_N & x_N & K_{N1} & \cdot & \cdot & 0 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ \\ P_1 \\ P_2 \\ \cdot \\ \cdot \\ P_N \end{Bmatrix} = \boldsymbol{C}\ \boldsymbol{P} \tag{8-20}$$

The interpolation to any point in the plane (x, y) is then achieved by evaluating $w(x, y)$ from Equation 8-17 at the desired points. This gives an overall equation of the form:

$$w_a = \begin{bmatrix} 1 & x_{1a} & y_{1a} & K_{1a,1} & K_{1a,2} & \cdots & K_{1a,n} \\ 1 & x_{2a} & y_{2a} & K_{2a,1} & K_{2a,2} & \cdots & K_{2a,n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot \\ 1 & x_{na} & y_{na} & K_{na,1} & K_{na,1^2} & \cdots & K_{na,n} \end{bmatrix} C^{-1} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ w_1 \\ w_2 \\ \cdot \\ w_N \end{Bmatrix} \tag{8-21}$$

Slopes of the aerodynamic panels, which are the negative of the slopes of the displacements, are also required. These can be determined by differentiating Equation 8-21 with respect to x:

$$\alpha_a = -\left\{\frac{\partial w}{\partial x}\right\}_a = -\begin{bmatrix} 0 & 1 & 0 & DK_{1a,1} & \cdots & DK_{1a,n} \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ 0 & 1 & 0 & DK_{na,1} & \cdots & DK_{na,n} \end{bmatrix} C^{-1} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ w_1 \\ \cdot \\ w_N \end{Bmatrix} \tag{8-22}$$

where

$$DK_{i,\ j} = \frac{\partial k_i\,(x_j, y_{i)}}{\partial x} = \left(\frac{x - x_i}{8\,\pi\,D}\right)\left(1 + nr_i^2\right) \tag{8-23}$$

8.3.2. Linear Spline

While linear splines may be solved by the three moment method, this technique does not work well for splines with torsion, rigid arms and attachment springs. The following outline is based on an analogy with the surface spline (see Section 17, Reference 1).

8.3.2.1. Linear Splines

The linear spline satisfies the equation:

$$EI\frac{d^4w}{dx^4} = q - \frac{dM}{dx} \tag{8-24}$$

where q is the applied load, and M is the applied moment. A symmetric fundamental solution for $x \neq 0$ is used for loads $q = P\,\delta(x)$, and an antisymmetric fundamental solution is used for moments. The solution for the general case is found by superimposing these fundamental solutions:

$$w(x) = a_o + a_1 x + \sum_{i=1}^{N}\left(\frac{M_i\,(x - x_i)\,|\,x - x_i|}{4EI} + \frac{P_i\,|\,x - x_i|^3}{12EI}\right) \tag{8-25}$$

$$\theta(x) = \frac{dw}{dx} = a_1 + \sum_{i=1}^{N} \left(\frac{M_i \mid x - x_i \mid}{2EI} + \frac{P_i (x - x_i) \mid x - x_i \mid}{4EI} \right) \tag{8-26}$$

These equations may be written in matrix form as:

$$\begin{Bmatrix} w(x) \\ \theta(x) \end{Bmatrix} = \begin{bmatrix} 1 & x & \frac{\mid x - x_1 \mid^3}{12EI} & \cdots & -\frac{(x - x_1) \mid x - x_1 \mid}{4EI} & \cdots \\ 0 & 1 & \frac{(x - x_1) \mid x - x_1 \mid}{4EI} & \cdots & -\frac{\mid x - x_1 \mid}{2EI} & \cdots \end{bmatrix} \begin{Bmatrix} a_o \\ a_1 \\ \hline P_1 \\ \cdots \\ P_N \\ \hline M_1 \\ \cdots \\ M_N \end{Bmatrix} \tag{8-27}$$

The unknowns, a, P_i, and M_i, may then be found from:

$$\begin{Bmatrix} 0 \\ 0 \\ \hline w_1 \\ \cdots \\ w_N \\ \hline \theta_1 \\ \cdots \\ \theta_N \end{Bmatrix} = \begin{bmatrix} 0 & \boldsymbol{R}_1^T & \boldsymbol{R}_2^T \\ \boldsymbol{R}_1 & \boldsymbol{A}_{11} & \boldsymbol{A}_{21}^T \\ \boldsymbol{R}_2 & \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \begin{Bmatrix} a_o \\ a_1 \\ \hline P_1 \\ \cdots \\ P_N \\ \hline M_1 \\ \cdots \\ M_N \end{Bmatrix} \tag{8-28}$$

where it is assumed that the x_i are monotonically increasing (i.e. $x_1 < x_2 < \ldots < x_N$), and:

$$\boldsymbol{R}_1^T = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_N \end{bmatrix} \tag{8-29a}$$

$$\boldsymbol{R}_2^T = \begin{bmatrix} 0 & 0 & \ldots & 0 \\ 1 & 1 & \ldots & 1 \end{bmatrix} \tag{8-29b}$$

$$A_{11} = \begin{bmatrix} 0 & \frac{(x_2 - x_1)^3}{12EI} & \cdots & \frac{(x_N - x_1)^3}{12EI} \\ \frac{(x_1 - x_2)^3}{12EI} & 0 & \cdots & \frac{(x_N - x_2)^3}{12EI} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{(x_1 - x_N)^3}{12EI} & \frac{(x_2 - x_N)^3}{12EI} & \cdots & 0 \end{bmatrix} \tag{8-29c}$$

$$A_{21} = \begin{bmatrix} 0 & -\frac{(x_2 - x_1)^2}{4EI} & \cdots & -\frac{(x_N - x_1)^2}{4EI} \\ \frac{(x_1 - x_2)^2}{4EI} & 0 & \cdots & -\frac{(x_N - x_2)^2}{4EI} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{(x_1 - x_N)^2}{4EI} & \frac{(x_2 - x_N)^2}{4EI} & \cdots & 0 \end{bmatrix} \tag{8-29d}$$

$$A_{22} = \begin{bmatrix} 0 & -\frac{(x_2 - x_1)}{2EI} & \cdots & -\frac{(x_N - x_1)}{2EI} \\ \frac{(x_1 - x_2)}{2EI} & 0 & \cdots & -\frac{(x_N - x_2)}{2EI} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{(x_1 - x_N)}{2EI} & \frac{(x_2 - x_N)}{2EI} & \cdots & 0 \end{bmatrix} \tag{8-29e}$$

8.3.2.2. Torsion Bars

In the case of torsion bars, the differential equation is:

$$GJ\frac{d^2\theta}{dx^2} = -T \tag{8-30}$$

for which the solution is:

$$\theta(x) = \left[1 \ , \ 0 \ , \ \frac{-|x_2 - x_1|}{2GJ} \ , \ \cdots \ , \ \frac{-|x_N - x_1|}{2GJ}\right] \begin{Bmatrix} a_o \\ T \end{Bmatrix} \tag{8-31}$$

$$\begin{Bmatrix} 0 \\ \hline \theta_1 \\ \cdots \\ \theta_N \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & 0 & \dfrac{-\left| x_2 - x_1 \right|}{2GJ} & \cdots & \dfrac{-\left| x_N - x_1 \right|}{2GJ} \\ 1 & \dfrac{-\left| x_1 - x_2 \right|}{2GJ} & 0 & \cdots & \dfrac{-\left| x_N - x_2 \right|}{2GJ} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \dfrac{-\left| x_1 - x_N \right|}{2GJ} & \dfrac{-\left| x_2 - x_N \right|}{2GJ} & \cdots & 0 \end{bmatrix} = \begin{Bmatrix} a_o \\ \hline T_1 \\ \cdots \\ T_N \end{Bmatrix} \tag{8-32}$$

8.3.3. Attachment of Splines with Elastic Springs

The change in the formulas to accomodate the springs for both surface and beam splines is straightforward. The spline deflection, given by Equations (8-19), (8-27) or (8-31) can be written:

$$\boldsymbol{u}_k(r) = \boldsymbol{R}(r)\,a + \boldsymbol{A}_j(r)\,\boldsymbol{P} \tag{8-33}$$

where $\boldsymbol{u}_k$ is the deflection of the spline and r may be a one- or two-dimensional argument. Including the equilibrium Equations (8-28) or (8-32) results in:

$$\boldsymbol{R}_i^T \boldsymbol{P} = \boldsymbol{0} \tag{8-34}$$

and

$$\boldsymbol{u}_k = \boldsymbol{R}_i\,\boldsymbol{a} + \boldsymbol{A}_{ij}\,\boldsymbol{P} \tag{8-35}$$

The structural deflection, $\boldsymbol{u}_g$, will differ from the spline deflection of the spring resulting in the forces:

$$\boldsymbol{P} = \boldsymbol{K}_s\left[\,\boldsymbol{u}_g - \boldsymbol{u}_k\,\right] \tag{8-36}$$

where the matrix, $\boldsymbol{K}_s$, has the spring constant, k, along its diagonal. These are nonzero (otherwise there would be no attachment and the grid point would be discarded) and thus the inverse of $\boldsymbol{K}_s$ is simply:

$$\boldsymbol{K}_s^{-1} = \begin{bmatrix} \dfrac{1}{k} & \cdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \cdots & \dfrac{1}{k} \end{bmatrix} \tag{8-37}$$

Eliminating u_k from Equations (8-35) and (8-36) results in:

$$u_g = R_i a + \left[A_{ij} + K_s^{-1} \right] P \tag{8-38}$$

Thus, all that is required to accommodate springs is to add the spring flexibilities to the diagonal of the spline influence coefficient matrix.

8.3.4. Rigid Arms on Splines

The linear splines used for geometry interpolation have rigid arms. Mathematically, these represent equations of constraint between the displacements and rotations at the spline end and attachment end. The constraint equations are used to transform the influence functions from the spline ends to influence functions at the attachment ends. The complete transformed influence functions are given in the following equations:

8.3.4.1. The *A* Matrix of Equation (8-33) for Surface Splines

$$\begin{Bmatrix} u_z \\ \theta_x \\ \theta_y \end{Bmatrix}_i = \begin{bmatrix} \dfrac{r_{ij}^2 \ln\,(r_{ij}^2)}{16\,\pi D} + \dfrac{\delta_{ij}}{k_z} \\ \dfrac{(y_i - y_j)\left(1 + \ln\,(r_{ij}^2\right)}{8\,\pi D} \\ \dfrac{(x_i - x_j)\left(1 + \ln\,(r_{ij}^2\right)}{8\,\pi D} \end{bmatrix} P_j \tag{8-39}$$

8.3.4.2. The *R* Matrix of Equation (8-33) for Surface and Linear Splines

$$\begin{Bmatrix} u_z \\ \theta_x \\ \theta_y \end{Bmatrix}_i = \begin{bmatrix} 1 & y_i & -x_i \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u_z \\ \theta_x \\ \theta_y \end{Bmatrix}_r \tag{8-40}$$

8.3.4.3. The *A* Matrix of Equation (8-33) for Linear Splines

$$\begin{Bmatrix} u_z \\ \theta_x \\ \theta_y \end{Bmatrix}_i = \begin{bmatrix} \frac{|y_i - y_j|}{12EI} - \frac{x_i x_j |y_i - y_j|}{2GJ} + \frac{\delta_{ij}}{k_z} & -\frac{|y_i - y_j|(y_i - y_j)}{4EI} & \frac{x_i |y_i - y_j|}{2GJ} \\ \frac{|y_i - y_j|(y_i - y_j)}{4EI} & -\frac{|y_i - y_j|}{2EI} + \frac{\delta_{ij}}{k_{\theta x}} & -\frac{(x_N - x_2)}{2EI} \\ \frac{x_j |y_i - y_j|}{2GJ} & 0 & -\frac{|y_i - y_j|}{2GJ} + \frac{\delta_{ij}}{k_{\theta y}} \end{bmatrix} \begin{Bmatrix} p_z \\ M_x \\ M_y \end{Bmatrix}_j \quad (8\text{-}41)$$

8.3.5. Equivalent Force Transfer

A second means of transferring loads from aerodynamic panels to the structure has been implemented for the frequently encountered case where no structural model exists for a particular aerodynamic component. The sketch of Figure 20 shows an example where the aerodynamic model contains a wing and horizontal tail surface while only the wing is modeled for the structural design task.

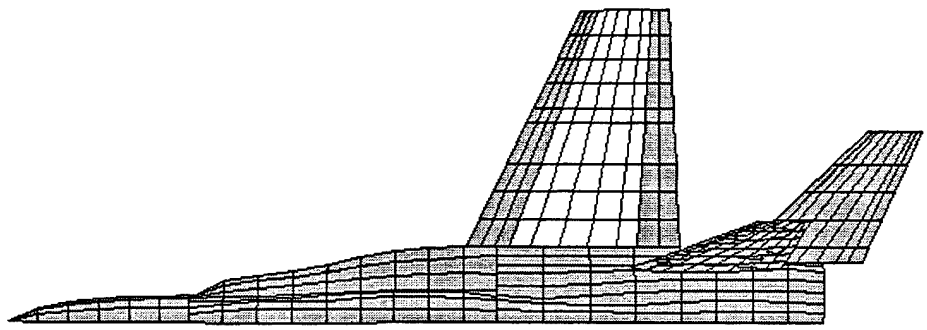


Figure 20. Application of ATTACH Option

This is done when only wing structural design is of interest, but the aerodynamic trim requires the determination of aerodynamic loads on the entire aircraft. The **ATTACH** bulk data entry of ASTROS permits the transfer of the loads from the aerodynamic panels to a specified grid in the structural model. This is done by a simple geometric transfer of the panel forces:

$$\boldsymbol{F}_R = \sum_{i=1}^{NBOX} \boldsymbol{F}_i$$

$$\boldsymbol{M}_R = \sum_{i=1}^{NBOX} \boldsymbol{R}_i \ \boldsymbol{F}_i \qquad (8\text{-}42)$$

where the R subscript refers to the structural grid and the i subscript identifies the individual aerodynamic box. The $\boldsymbol{R}$ matrix to compute the equivalent moments is simply:

$$\boldsymbol{R}_i = \begin{bmatrix} 0 & -\left(z_i - z_R\right) & \left(y_i - y_R\right) \\ \left(z_i - z_r\right) & 0 & -\left(x_i - x_R\right) \\ -\left(y_i - y_R\right) & \left(x_i - x_R\right) & 0 \end{bmatrix} \qquad (8\text{-}43)$$

The transformations of Equations 8-42 are integrated with the spline transformation of Equation 8-46 or 8-47 so that every aerodynamic load is transferred to the structure.

8.3.6. Use of Splines

The preface modules of ASTROS use the relations of Equations 8-21 and 8-22 to create the required spline matrices. Steady and unsteady aerodynamics have different requirements and therefore different splines are created. For unsteady aerodynamics, displacements and slopes are required at the aerodynamic grid points so that the spline matrix interleaves results of Equations 8-21 and 8-22 to give a matrix with the number of rows equal to two times the number of lifting surface panels. (Surface splines are not used to compute displacements on body panels.) Symbolically

$$\boldsymbol{w}_a = \boldsymbol{UG}\, \boldsymbol{w}_s \qquad (8\text{-}44)$$

where the a subscript refers to displacements and slopes at the aerodynamic grid points and the s subscripts refer to structural displacements. Conditions of virtual work can be applied to derive the fact that the transpose of the $\boldsymbol{UG}$ matrix relates forces in the two sets:

$$\boldsymbol{F}_s = \boldsymbol{UG}^T \boldsymbol{F}_a \qquad (8\text{-}45)$$

where the $\boldsymbol{F}_a$ vectors contain forces and moments at the aerodynamic panel and $\boldsymbol{F}_s$ contains the out-of-plane forces at the structural grid points.

For steady aerodynamics, ASTROS has generated ***AIC*** matrices that relate forces on aerodynamic panels due to slopes at the panel. Two separate matrices are generated in this case. The first utilizes Equation 8-22 to compute aerodynamic slopes:

$$\alpha_a = \boldsymbol{GS}\ \boldsymbol{w}_s \tag{8-46}$$

while the second uses Equation 8-21 to compute structural forces:

$$\boldsymbol{F}_s = \boldsymbol{G}\ \boldsymbol{F}_a \tag{8-47}$$

9. STATIC AEROELASTIC ANALYSIS

The static aeroelastic analysis features in ASTROS provide the capability to analyze and design linear structures in the presence of steady aerodynamic loading. This provides the ASTROS user with a self-contained capability to compute loads experienced by a maneuvering aircraft and to redesign the structure based on these loads. The capabilities available for steady aerodynamics design include specifying limits on (1) the allowable stress or strain response due to a specified trimmed maneuver, (2) the flexible to rigid ratio of the aircraft's lift curve slope, (3) the flexible roll control effectiveness of any antisymmetric control surface and (4) the values of the flexible stability derivatives and trim parameters. This section first defines the basic equations used for static aeroelastic analyses and then contains individual subsections for each of the listed design aspects.

9.1. MATRIX EQUATIONS FOR STATIC AEROELASTIC ANALYSIS

The equations for static analysis given in Subsection 6.1 can be easily adapted for steady aerodynamic analysis. In fact, Equations 6-1 through 6-9 are equally applicable to static and steady aerodynamic analysis, since there is no interaction between mass, stiffness and aerodynamic terms in the reduction to the *f-set*. Reduction of the aerodynamic forces to the *a-set* does require coupling with the stiffness matrix so that it is at the *f-set* that the aerodynamic and structural stiffnesses are joined. The spline matrices of Equations 8-25 and 8-26 do require reduction to the *f-set* and these reductions are similar to the reduction of the applied loads given in Equations 6-5 and 6-9:

$$G_{jn}^T = \overline{G}_{jn}^T + T_{MN}^T G_{jm}^T$$

$$\left[G_s \right]_{jn}^T = \left[\overline{G}_s \right]_{jn}^T + T_{MN}^T \left[G_s \right]_{jm}^T \tag{9-1}$$

$$G_{jf}^T = \overline{G}_{jf}^T$$

$$\left[G_s \right]_{jf}^T = \left[\overline{G}_s \right]_{jf}^T \tag{9-2}$$

where the transposed matrices are used for convenience and the assumption is made that there are no enforced displacements. The j subscript denotes the panels in the aerodynamic model.

The aerodynamic forces and influence coefficients of Subsection 8.1.2 are then applied to the structure through the following splining relation:

$$P_f^a = \overline{q}\, G_{jf}^T \; \textit{AIRFRC} \tag{9-3}$$

$$AICS_{ff} = \bar{q}\, G_{jf}^{T}\, AIC \left[G_s \right]_{jf} \tag{9-4}$$

where

P_f^a Unit aerodynamic loads matrix

$AICS$ Aerodynamic influence coefficient matrix

$\bar{q}$ Dynamic pressure

The aerodynamic terms are added to the structural terms to give:

$$\left[K_{ff} - AICS_{ff} \right] u_f + M_{ff} \ddot{u}_f = P_f^a\, \delta \tag{9-5}$$

where δ is a vector of configuration parameters, such as angle of attack and elevator angle. The δ vector is explicitly defined for the symmetric and antisymmetric cases in Subsections 9.3 and 9.4, respectively.

It is convenient to define a new matrix which is the difference of the structural and the aerodynamic stiffnesses:

$$K_{ff}^a = K_{ff} - AICS_{ff} \tag{9-6}$$

The reduction of Equation 9-5 to the *l-set* and *r-set* is very similar to the formulation of Equations 6-10 through 6-21 and this similarity is drawn on here.

Dynamic reduction of the steady aeroelastic equations is not supported. Guyan reduction may be performed, and, if so, the relationships of Equations 6-11 and 6-12 are modified to account for the aerodynamic stiffness:

$$\ddot{u}_o = -\left[K_{oo}^a \right]^{-1} K_{oa}^a \ddot{u}_a = G_o^a \ddot{u}_a \tag{9-7}$$

$$u_o = \left[K_{oo}^a \right]^{-1} P_o^a \delta - \left[K_{oo}^a \right]^{-1} K_{oa}^a u_a \tag{9-8}$$

$$K_{aa}^a u_a + M_{aa} \ddot{u}_a = P_a^a \delta \tag{9-9}$$

where

$$K_{aa}^a = \bar{K}_{aa}^a - K_{ao}^a G_o^a$$

$$P_a^a = \bar{P}_a^a - K_{ao}^a \left[K_{oo}^a \right]^{-1} P_o^a \tag{9-10}$$

$$M_{aa} = \bar{M}_{aa} + M_{oa} G_o^a + \left[G_o^a \right]^T M_{oa}^T + \left[G_o^a \right]^T M_{oo} G_o^a$$

Note that since the K^a matrix is not symmetric, it is necessary to retain both the K^a_{oa} and K^a_{ao} portion of this matrix for subsequent operations.

Equation 9-9 can be partitioned into *r-set* and *l-set* degrees of freedom:

$$\begin{bmatrix} K^a_{ll} & K^a_{lr} \\ K^a_{rl} & K^a_{rr} \end{bmatrix} \begin{Bmatrix} u_l \\ u_r \end{Bmatrix} + \begin{bmatrix} M_{ll} & M_{lr} \\ M_{rl} & M_{rr} \end{bmatrix} \begin{Bmatrix} \ddot{u}_l \\ \ddot{u}_r \end{Bmatrix} = \begin{Bmatrix} P^a_l \\ P^a_r \end{Bmatrix} \delta \tag{9-11}$$

As in the inertia relief formulation, $\ddot{u}_l$ and $\ddot{u}_r$ are related through the equation.

$$\ddot{u}_l = D\ \ddot{u}_r \tag{6-18}$$

Note that elastic accelerations are not treated in this formulation.

Just as was discussed in statics with inertial relief, the *r-set* contains degrees of freedom equal in number to the number of rigid body modes in the structure. Static aeroelastic analysis differs from statics with inertial relief in the way the *r-set* displacements are calculated. In statics, the *r-set* displacements are arbitrarily set to zero. For static aeroelasticity, the *a-set* displacements are determined by requiring that these elastic deformations be orthogonal to the rigid body motions. In terms of internal loads, these two approaches are equivalent since only the elastic deformations produce these loads.

The orthogonality condition has been imposed to produce aerodynamic stability derivatives that are independent of the degrees of freedom included in the *r-set*. The static aeroelastic capability used in MSC/NASTRAN (Reference 16) provided the basis for this concept. A consequence of this revised formulation is that inertia relief loads must always be included in the ASTROS static aeroelastic analysis whereas statics can solve for elastic deformations for free bodies without considering the mass terms. This is in consequence of the orthogonality condition, which requires the mass matrix in its specification.

The orthogonality constraint between elastic deformations and rigid body motions is specified by:

$$\begin{bmatrix} D^T I \end{bmatrix} \begin{bmatrix} M_{ll} & M_{lr} \\ M_{rl} & M_{rr} \end{bmatrix} \begin{Bmatrix} u_l \\ u_r \end{Bmatrix} = 0 \tag{9-12}$$

If Equation 6-18 and the orthogonality condition of Equation 9-12 are inserted into Equation 9-11, the resulting equation is

$$\begin{bmatrix} K^a_{ll} & K^a_{lr} & M_{ll}D + M_{lr} \\ K^a_{rl} & K^a_{rr} & M_{rl}D + M_{rr} \\ D^T M_{ll} + M_{rl} & D^T M_{lr} + M_{rr} & 0 \end{bmatrix} \begin{Bmatrix} u_l \\ u_r \\ \ddot{u}_r \end{Bmatrix} = \begin{Bmatrix} P^a_l \\ P^a_r \\ 0 \end{Bmatrix} \delta \tag{9-13}$$

These equations can be solved in a variety of ways, with a particular algorithm entailing multiplying the first row of Equation 9-13 by $\boldsymbol{D}^T$ and adding it to the second row. This new second row is interchanged with the third equation to give the following system:

$$\begin{bmatrix} \boldsymbol{K}_{ll}^a & \boldsymbol{K}_{lr}^a & \boldsymbol{M}_{ll}\boldsymbol{D}+\boldsymbol{M}_{lr} \\ \boldsymbol{D}^T\boldsymbol{M}_{ll}+\boldsymbol{M}_{rl} & \boldsymbol{D}^T\boldsymbol{M}_{lr}+\boldsymbol{M}_{rr} & \boldsymbol{0} \\ \boldsymbol{D}^T\boldsymbol{K}_{ll}^a+\boldsymbol{K}_{rl}^a & \boldsymbol{D}^T\boldsymbol{K}_{lr}^a+\boldsymbol{K}_{rr}^a & \boldsymbol{m}_r \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_l \\ \boldsymbol{u}_r \\ \ddot{\boldsymbol{u}}_r \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{P}_l^a \\ \boldsymbol{0} \\ \boldsymbol{D}^T\boldsymbol{P}_l^a+\boldsymbol{P}_r^a \end{Bmatrix} \delta \tag{9-14}$$

where $\boldsymbol{m}_r$ is the reduced mass matrix of Equation 6-21 and, unlike the static analysis equation of Equation 6-20, the R_{31} and R_{32} terms of Equation 9-14 have nonzero contributions from the aerodynamic corrections.

Equation 9-14 is redefined in order to simplify the notation based on the partitions given in the equation:

$$\begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} & \boldsymbol{R}_{13} \\ \boldsymbol{R}_{21} & \boldsymbol{R}_{22} & \boldsymbol{R}_{23} \\ \boldsymbol{R}_{31} & \boldsymbol{R}_{32} & \boldsymbol{R}_{33} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_l \\ \boldsymbol{u}_r \\ \ddot{\boldsymbol{u}}_r \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{P}_l^a \\ \boldsymbol{0} \\ \boldsymbol{D}^T\boldsymbol{P}_l^a + \boldsymbol{P}_r^a \end{Bmatrix} \delta \tag{9-15}$$

Then, one possible solution algorithm involves merging the matrices into

$$\begin{bmatrix} \overline{\boldsymbol{K}}_{11} & \overline{\boldsymbol{K}}_{12} \\ \overline{\boldsymbol{K}}_{21} & \overline{\boldsymbol{K}}_{22} \end{bmatrix} \begin{Bmatrix} \overline{\boldsymbol{u}}_1 \\ \overline{\boldsymbol{u}}_2 \end{Bmatrix} = \begin{Bmatrix} \overline{\boldsymbol{P}}_1 \\ \overline{\boldsymbol{P}}_2 \end{Bmatrix} \delta \tag{9-15a}$$

$$\overline{\boldsymbol{K}}_{11} = \begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \boldsymbol{R}_{21} & \boldsymbol{R}_{22} \end{bmatrix} \tag{9-15b}$$

$$\overline{\boldsymbol{K}}_{12} = \boldsymbol{R}_{13} \tag{9-15c}$$

$$\overline{\boldsymbol{K}}_{21} = \begin{bmatrix} \boldsymbol{R}_{31} & \boldsymbol{R}_{32} \end{bmatrix} \tag{9-15d}$$

$$\overline{\boldsymbol{K}}_{22} = \boldsymbol{R}_{33} \tag{9-15e}$$

$$\overline{\boldsymbol{u}}_1 = \boldsymbol{u}_a = \begin{Bmatrix} \boldsymbol{u}_l \\ \boldsymbol{u}_r \end{Bmatrix} \tag{9-15f}$$

$$\overline{\boldsymbol{u}}_2 = \ddot{\boldsymbol{u}}_r \tag{9-15g}$$

$$\overline{P}_1 = \begin{Bmatrix} P_l^a \\ 0 \end{Bmatrix} \tag{9-15h}$$

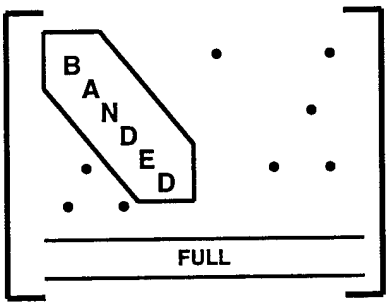
$$\overline{P}_2 = D^T P_l^a + P_r^a \tag{9-15i}$$

The first row of Equation 9-15a can be solved for $\overline{u}_1$ in terms of δ and $\overline{u}_2$ to give

$$\overline{u}_1 = \overline{K}_{11}^{-1} \left[\overline{P}_1 \ \delta - \overline{K}_{12} \overline{u}_2 \right] \tag{9-16}$$

Equation 9-16 requires the decomposition of the $\overline{K}_{11}$ matrix.

While the R_{11} partition of the $\overline{K}_{11}$ matrix is asymmetric, it is also very sparse and reasonably well banded (assuming that the structural stiffness, K_{aa} is banded) and the asymmetries and unbandedness are only associated with those degrees of freedom to which the aeroelastic correction matrix have been splined. Unfortunately, the orthogonality criterion introduces one row for each support degree of freedom that is fully coupled. Thus the $\overline{K}_{11}$ matrix has a topology like:



This topology results in very poor performance in the solution of 9-16. As a result the trim equations 9-15 are solved in a slightly different order to allow the R_{11} matrix to be decomposed rather than the $\overline{K}_{11}$ matrix.

Solving for u_l from the first row of:

$$u_l = R_{11}^{-1} \left[P_l^a \ \delta - R_{12} u_r - R_{13} \ddot{u}_r \right] \tag{9-17}$$

and substituting for u_l in the second and third rows, we obtain the trim equations in the same form as Equation 9-15a, except that u_1 contains only the rigid body displacements. The flexible displacements must then be computed from equation 9-17. This is detailed below.

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \begin{Bmatrix} P_1 \\ P_2 \end{Bmatrix} \{\delta\} \tag{9-17a}$$

$$\boldsymbol{K}_{11} = \boldsymbol{R}_{22} - \boldsymbol{R}_{21}\,\boldsymbol{R}_{11}^{-1}\,\boldsymbol{R}_{12} \tag{9-17b}$$

$$\boldsymbol{K}_{12} = \boldsymbol{R}_{23} - \boldsymbol{R}_{21}\,\boldsymbol{R}_{11}^{-1}\,\boldsymbol{R}_{13} \tag{9-17c}$$

$$\boldsymbol{K}_{21} = \boldsymbol{R}_{32} - \boldsymbol{R}_{31}\,\boldsymbol{R}_{11}^{-1}\,\boldsymbol{R}_{12} \tag{9-17d}$$

$$\boldsymbol{K}_{22} = \boldsymbol{R}_{33} - \boldsymbol{R}_{31}\,\boldsymbol{R}_{11}^{-1}\,\boldsymbol{R}_{13} \tag{9-17e}$$

$$\boldsymbol{P}_1 = -\,\boldsymbol{R}_{21}\boldsymbol{R}_{11}^{-1}\,\boldsymbol{P}\boldsymbol{A}_l \tag{9-17f}$$

$$\boldsymbol{P}_2 = \boldsymbol{D}^T\boldsymbol{P}_l^a + \boldsymbol{P}_r^a - \boldsymbol{R}_{31}\boldsymbol{R}_{11}^{-1}\,\boldsymbol{P}_l^a \tag{9-17g}$$

$$\boldsymbol{u}_1 = \boldsymbol{u}_r \tag{9-17h}$$

$$\boldsymbol{u}_2 = \ddot{\boldsymbol{u}}_r \tag{9-17i}$$

As in the previous formulation

$$\boldsymbol{u}_1 = \boldsymbol{K}_{11}^{-1}\left[\boldsymbol{P}_1\,\delta - \boldsymbol{K}_{12}\,\boldsymbol{u}_2\right] \tag{9-18}$$

and upon substitution into 9-17a:

$$\left[\boldsymbol{K}_{22} - \boldsymbol{K}_{21}\boldsymbol{K}_{11}^{-1}\boldsymbol{K}_{12}\right]\boldsymbol{u}_2 = \left[\boldsymbol{P}_2 - \boldsymbol{K}_{21}\boldsymbol{K}_{11}^{-1}\boldsymbol{P}_1\right]\delta \tag{9-19}$$

While the formulation still involves an inversion of $\boldsymbol{K}_{11}$, (in addition to $\boldsymbol{R}_{11}$) that matrix is now an r x r matrix rather than $(a+r)$ x $(a+r)$. The inversion of $\boldsymbol{R}_{11}$ can be performed relatively efficiently.

Equation 9-19 is the basic equation for static aeroelastic analysis. There is one equation in 9-19 for each rigid body degree of freedom. In general, then, *nr* unknowns can be determined from these equations. The $\boldsymbol{u}_2$ vector is the vector of structural accelerations at the support point (in the global coordinate system) and the δ vector is a vector of trim parameters. The user is free to pick any number of **fixed** values of $\boldsymbol{u}_2$ or δ rows and exactly *nr* **free** values to be determined by the solution of Equation 9-19. Symmetric and antisymmetric analyses limit the set of accelerations and configuration parameters. If Equation 9-19 is partitioned into free and known values:

$$\boldsymbol{L}\,\boldsymbol{u}_2 = \boldsymbol{R}\,\delta \tag{9-20}$$

where

$$\boldsymbol{L} = \boldsymbol{K}_{22} - \boldsymbol{K}_{21}\boldsymbol{K}_{11}^{-1}\boldsymbol{K}_{12} \tag{9-20a}$$

$$\boldsymbol{R} = \boldsymbol{P}_2 - \boldsymbol{K}_{21}\boldsymbol{K}_{11}^{-1}\boldsymbol{P}_1 \tag{9-20b}$$

partitioning yields:

$$\begin{bmatrix} L_{ff} & L_{fk} \\ L_{kf} & L_{kk} \end{bmatrix} \begin{Bmatrix} u_{2f} \\ u_{2k} \end{Bmatrix} = \begin{bmatrix} R_{fu} & R_{fs} \\ R_{ku} & R_{ks} \end{bmatrix} \begin{Bmatrix} \delta_u \\ \delta_s \end{Bmatrix} \tag{9-21}$$

where the f and u subscripts denote **free** (or **unknown**) values and the k and s subscripts denote **known** (or **set**) values of accelerations and trim parameters, respectively. Rearranging to place free values at the left:

$$\begin{bmatrix} L_{ff} & -R_{fu} \\ L_{kf} & -R_{ku} \end{bmatrix} \begin{Bmatrix} u_{2f} \\ \delta_u \end{Bmatrix} = \begin{bmatrix} -L_{fk} & R_{fs} \\ -L_{kk} & R_{ks} \end{bmatrix} \begin{Bmatrix} u_{2k} \\ \delta_s \end{Bmatrix} \tag{9-22}$$

The set of values that can participate in Equation 9-22 is a function of ASTROS and the user's model. There are six rigid body accelerations, which in ASTROS have been given the names

$$u_2 \in \left\{ \begin{array}{l} \texttt{NX} \\ \texttt{NY} \\ \texttt{NZ} \\ \texttt{PACCEL} \\ \texttt{QACCEL} \\ \texttt{RACCEL} \end{array} \right\}$$

The δ vector has a number of predefined components and the user can add components by defining control surfaces. Thus, δ can be viewed as:

$$\delta \in \left\{ \begin{array}{l} \texttt{THKCAM} - \text{thickness \& camber} \\ \texttt{ALPHA} - \text{angle of attack} \\ \texttt{BETA} - \text{yaw angle} \\ \texttt{PRATE} - \text{rollrate} \\ \texttt{QRATE} - \text{pitchrate} \\ \texttt{RRATE} - \text{yawrate} \\ \{\delta_{sym}\} - \text{symmetric surfaces} \\ \{\delta_{anti}\} - \text{antisymmetric surfaces} \end{array} \right\}$$

Notice that **GAMMA**, the roll angle, is not available since it can produce no forces using the ASTROS aerodynamics method. Subsections 9.3 and 9.4 discuss particular applications of this equation for the symmetric and antisymmetric cases.

9.2. STABILITY DERIVATIVES

Since the aeroelastic stability derivatives form an important set of data used to generate design constraints, ASTROS computes stability derivatives for each trim analysis that is performed. Although the stability derivatives are, using the ASTROS aerodynamics, a function only of Mach number and

dynamic pressure, user input effectiveness values can cause the stability derivatives to vary from case to case.

Equation 9-19 can be used to compute the stability derivatives by realizing that, using an identity matrix for δ, the resulting $\boldsymbol{u}_2$ terms are the accelerations of the flexible vehicle due to unit parameters. Using the $\boldsymbol{m}_r$ matrix, the forces and moments can then be found.

$$\boldsymbol{F} = \boldsymbol{m}_r \left\{ \left[\boldsymbol{K}_{22} - \boldsymbol{K}_{21} \boldsymbol{K}_{11}^{-1} \boldsymbol{K}_{12} \right]^{-1} \left[\boldsymbol{P}_2 - \boldsymbol{K}_{21} \boldsymbol{K}_{11}^{-1} \boldsymbol{P}_1 \right] \right\} \tag{9-23}$$

$\boldsymbol{F}$ has one row for each supported degree of freedom and one column for each configuration parameter. Normalizing $\boldsymbol{F}$ to a nondimensional derivative must account for forces and moments and handle rate parameters as enumerated below:

$$\boldsymbol{F} \in \begin{Bmatrix} \boldsymbol{F}_x \\ \boldsymbol{F}_y \\ \boldsymbol{F}_z \\ \boldsymbol{M}_x \\ \boldsymbol{M}_y \\ \boldsymbol{M}_z \end{Bmatrix} = \begin{Bmatrix} \text{THRUST/DRAG} \\ \text{SIDE FORCE} \\ \text{LIFT} \\ \text{ROLL MOMENT} \\ \text{PITCH MOMENT} \\ \text{YAW MOMENT} \end{Bmatrix}$$

for forces, the nondimensional stability derivatives are ℓ

CONTROL SURFACES	RATE PARAMETERS	
$C_D = \dfrac{2\boldsymbol{F}_x}{\bar{q} S}$	$C_D = \dfrac{4\boldsymbol{F}_x}{\bar{q} S c}$	(9-24a)
$C_S = \dfrac{2\boldsymbol{F}_y}{\bar{q} S}$	$C_S = \dfrac{4\boldsymbol{F}_y}{\bar{q} S b}$	(9-24b)
$C_L = \dfrac{2\boldsymbol{F}_z}{\bar{q} S}$	$C_L = \dfrac{4\boldsymbol{F}_z}{\bar{q} S c}$	(9-24c)
$C_\ell = \dfrac{2\boldsymbol{M}_x}{\bar{q} S b}$	$C_\ell = \dfrac{4\boldsymbol{M}_x}{\bar{q} S b^2}$	(9-24d)
$C_m = \dfrac{2\boldsymbol{M}_y}{\bar{q} S c}$	$C_m = \dfrac{4\boldsymbol{M}_y}{\bar{q} S c^2}$	(9-24e)
$C_y = \dfrac{2\boldsymbol{M}_z}{\bar{q} S b}$	$C_y = \dfrac{4\boldsymbol{M}_z}{\bar{q} S b^2}$	(9-24f)

9.2.1. Restrained and Unrestrained Stability Derivatives

The F vector of Equation 9-23 is derived using the orthogonality criterion of Equation 9-12. This produces forces that are independent of the *r-set*. It also includes terms associated with inertia relief effects. The resultant stability derivatives are termed **unrestrained** since they represent the value associated with the free aircraft. If the orthogonality criterion is removed from Equation 9-14 and the trim equations are solved again using the standard $u_r \equiv 0$ to remove the singularities we obtain:

$$\begin{bmatrix} R_{11} & R_{13} \\ R_{31} & R_{33} \end{bmatrix} \begin{Bmatrix} u_l^R \\ \ddot{u}_r^R \end{Bmatrix} = \begin{Bmatrix} P_l^a \\ D^T P_l^a + P_r^a \end{Bmatrix} \delta \tag{9-25}$$

using the first row to eliminate u_l, we obtain our restrained trim equations:

$$u_l^R = R_{11}^{-1} \left[P_l^a \delta - R_{13} \ddot{u}_r^R \right] \tag{9-26a}$$

$$\left[R_{33} - R_{31} R_{11}^{-1} R_{13} \right] \ddot{u}_r^R = \left[D^T P_l^a + P_r^a - R_{31} R_{11}^{-1} P_l^a \right] \delta \tag{9-26b}$$

Again using the method of computing the accelerations due to unit δ values and premultiplying by m_r to obtain forces, we can compute the restrained values of F, denoted F_R.

$$F_R = m_r \left[R_{33} - R_{31} R_{11}^{-1} R_{13} \right]^{-1} \left[D^T P_l^a + P_r^a - R_{31} R_{11}^{-1} P_l^a \right] \tag{9-27}$$

by inspection, we can convert Equation 9-27 into K_{ij} notation:

$$F_R = m_r K_{22}^{-1} P_2 \tag{9-28}$$

compared with the unrestrained values

$$F = m_r \left[K_{22} - K_{21} K_{11}^{-1} K_{12} \right]^{-1} \left[P_2 - K_{21} K_{11}^{-1} P_1 \right] \tag{9-23}$$

Given one set of derivatives, the other set can be computed:

$$F_R = F + m_r K_{22}^{-1} K_{21} K_{11}^{-1} P_1 - m_r K_{22}^{-1} K_{21} K_{11}^{-1} K_{12} \ddot{u}_r \tag{9-29}$$

In this instance, the $\ddot{u}_r$ values are those associated with the unrestrained accelerations due to unit parameters. A similar expression can be obtained using the restrained accelerations.

These restrained coefficients, F_R , include inertia relief effects, however. These effects appear in the K_{22} matrix.

$$K_{22} = R_{33} - R_{31}\, R_{11}^{-1}\, R_{13} = m_r - \text{inertia relief} \tag{9-17e}$$

Since the calculation method multiplies K_{22}^{-1} by m_r, one can see that the restrained stability derivatives without inertia relief can be computed as:

$$F_{R_{NI}} = P_2 = D^T\, P_l^a + P_r^a - R_{31}\, R_{11}^{-1}\, P_l^a \tag{9-30}$$

Finally, one can revisit the unrestrained stability derivatives and separate the inertia relief components from the overall stability derivative. To do so we must examine Equation 9-23 and look at only the applied forces and omitting any force modifier associated with $\ddot{u}_r$ we then obtain:

$$F_{NI} = P_2 - K_{21}\, K_{11}^{-1}\, P_1 \tag{9-31}$$

The inertia relief effect can be derived from 9-17a (expanded) as shown: (the shaded terms are the inertia relief modifiers)

$$\begin{bmatrix} \left[R_{22} - R_{21}\, R_{11}^{-1}\, R_{12}\right] & -\left[R_{21}\, R_{11}^{-1}\, R_{13}\right] \\ \left[R_{32} - R_{31}\, R_{11}^{-1}\, R_{12}\right] & \left[R_{33} - R_{31}\, R_{11}^{-1}\, R_{13}\right] \end{bmatrix} \begin{Bmatrix} u_r \\ \ddot{u}_r \end{Bmatrix} = \begin{Bmatrix} -R_{21}\, R_{11}^{-1}\, P_l^a \\ D^T P_l^a + P_r^a - R_{31}\, R_{11}^{-1}\, P_l^a \end{Bmatrix} \delta \tag{9-32}$$

$$F_{inertia} = +\left[R_{21}\, R_{11}^{-1}\, R_{13} + R_{31}\, R_{11}^{-1}\, R_{13}\right] \ddot{u}_r \tag{9-33a}$$

$$= \left[-K_{12} - K_{22} + R_{33}\right] \ddot{u}_r \tag{9-33b}$$

Where $\ddot{u}_r$ are the accelerations due to unit parameters from the unrestrained formulation:

$$\ddot{u}_r = \left[K_{22} - K_{21}\, K_{11}^{-1}\, K_{12}\right]^{-1} \left[P_2 - K_{21}\, K_{11}^{-1}\, P_1\right] \tag{9-34}$$

In summary, there are four varieties of stability derivatives that can be examined. In ASTROS, they are each derived from a dimensional force vector using the nondimensionalizing Equation 9-24. The force vectors themselves are:

Unrestrained (Orthogonality and Inertia Relief) :	F	from Equation 9-23.
Restrained (Orthogonality, no Inertia Relief) :	F_{NI}	from Equation 9-31.
Supported (no Orthogonality, but Inertia Relief) :	F_R	from Equation 9-28.
Fixed (neither Orthogonality nor Inertia Relief) :	$F_{R_{NI}}$	from Equation 9-30.

In ASTROS, the values of certain constraints use the unrestrained coefficients while others use the restrained values. The supported and fixed values are not used since their value in strongly controlled by the location of the support point.

9.3. SYMMETRIC ANALYSES

Symmetric steady aerodynamic analyses are applied in ASTROS for longitudinal trim and subsequent stress analysis and for analysis and design of an aircraft's symmetric aeroelastic and trim parameters and aeroelastic stability derivatives. For symmetric analyses, the δ vector has up to three predefined components and any number of user parameters and the u_2 vector can have up to three predetermined components.:

$$\delta \in \left\{ \begin{array}{l} \mathbf{THKCAM} \\ \mathbf{ALPHA} \\ \mathbf{QRATE} \\ \{\delta_{sym}\} \end{array} \right\}$$

$$u_2 \in \left\{ \begin{array}{c} \mathbf{NX} \\ \mathbf{NZ} \\ \mathbf{QACCEL} \end{array} \right\}$$

Thickness and camber effects refer to the airloads produced when the other members are zero and can be thought of as giving zero angle-of-attack effects. The value of this term is typically 1.0. The pitch control surfaces govern the motion of the aerodynamic panels that trim the pitching moment of the aircraft. These could represent an elevator or an all moving stabilizer on a canard or tail surface. These terms are designated as δ_{sym}. Pitch rate (**QRATE**) is designated as q, while the angle of attack parameter (**ALPHA**) is denoted by α.

9.3.1. Trim Analysis

For the trim analysis, Equation 9-22 is solved for u_{2f} and δ_u. The u_{2f} and δ_u vectors have between them as many terms as there are in the *r-set* (*nr*). u_2 and δ are then obtained by merging the known values with the computed values.

Given the values for the u_2 and δ vectors, the recovery of the elastic deformations is straightforward. The u_1 vector of Equation 9-17a is the u_r vector of Equation 9-14. Flexible deformations are then recovered using Equation 9-17 while the *l-set* accelerations are computed using Equation 6-18. Further recovery of the omitted degrees of freedom and the single and multiple point constraints proceeds as detailed in Subsection 6.1. One difference from that formulation is in how loads applied to omitted degrees of freedom affect the omitted displacements. These aerodynamic loads are computed using

$$P_o = P_o^a \, \delta \tag{9-35}$$

where P_o^a is the matrix of rigid aerodynamic loads on the omitted degrees of freedom and δ is the vector of trim parameters determined during the trim process. These omitted loads are then used to recover omitted displacements in the standard fashion:

$$u_o^o = \left[K_{oo}^a \right]^{-1} \left[P_o - \left[M_{oo} G_o^a + M_{oa} \right] u_a \right] \tag{9-36}$$

$$u_o = G_o^a u_a + u_o^o \tag{9-37a}$$

$$\ddot{u}_o = G_o^a \ddot{u}_a \tag{9-37b}$$

where G_o^a is defined in Equation 9-7.

Recovery of accelerations and displacements in the *f-set* and *g-set* proceeds normally. Given the displacements in the *g-set*, displacement constraints can be calculated and Equation 6-28 can be used to recover the components used in computing strength constraints.

9.3.2. Lift Effectiveness Constraint

The lift effectiveness constraint in ASTROS places bounds on the ratio of the flexible to rigid lift curve slope of the aircraft

$$\varepsilon_{\min} \le \frac{C_{L_{\alpha_f}}}{C_{L_{\alpha_R}}} \le \varepsilon_{\max} \tag{4-27}$$

Subsection 4.5.2.4 defines the terms used in this equation.

Equation 9-23 outlines the basic approach to evaluate this constraint. Conceptually, the flexible lift curve slope is obtained by setting the term corresponding to the angle of attack in the δ vector to unity and the remaining terms in the vector to zero and then determining the resulting values of u_2. These are the accelerations of the aircraft and, when multiplied by the matrix m_r, give the force and moment acting on the structure. These can then be nondimensionalized to stability derivatives with the force term translating to the lift derivative. In mathematical terms:

$$\frac{\bar{q}S}{2} \begin{Bmatrix} C_{L_{\alpha_f}} \\ c\, C_{m_{\alpha_f}} \end{Bmatrix} = m_r \left[K_{22} - K_{21} K_{11}^{-1} K_{12} \right]^{-1} \left[P_2 - K_{21} K_{11}^{-1} P_1 \right] \quad \delta_\alpha = F \delta_\alpha \tag{9-38}$$

Where δ_α is the configuration vector δ with a unit value of the angle of attack, S is the wing reference area, c is the wing reference chord and the C_L value is associated with the vertical translation DOF of the supported point, while C_m is associated with the pitch rotation DOF. The factor of two on the left-hand side of Equation 9-38 is due to the fact that the right-hand side equations account for only one side of the aircraft.

Rigid stability derivatives are determined from a less complex matrix equation:

Table 11. Lift Effectiveness Constraint Coefficients

SIGN OF ε_{req}	CONSTRAINT TYPE			
	UPPER		LOWER	
	a	b	a	b
POS	- 1.0	$\frac{1}{\varepsilon_{req}}$	1.0	$-\frac{1}{\varepsilon_{req}}$
NEG	1.0	$-\frac{1}{\varepsilon_{req}}$	- 1.0	$\frac{1}{\varepsilon_{req}}$
ZERO	0.0	1.0	1.0	- 1.0

$$\frac{\bar{q}S}{2}\begin{Bmatrix} C_{L_{\alpha_R}} \\ c\,C_{m_{\alpha_R}} \end{Bmatrix} = \boldsymbol{P}_2\,\delta_\alpha = F_{R_{NI}}\,\delta_\alpha \tag{9-39}$$

The lift effectiveness constraint is calculated using

$$g = a + b\,\varepsilon \tag{9-40}$$

where ε is the flexible to rigid ratio of Equation 9-38. The a and b coefficients are listed in Table 11 for upper and lower bound constraints and required effectiveness values (ε_{req}) that are positive, negative or zero.

Specification of upper bound limits on the effectiveness and negative and zero values of the required effectiveness have been included for completeness. It is anticipated that these particular features will rarely be used.

9.3.3. Flexible Stability Derivatives

A general extension of Equation 9-38 can be used to compute the flexible stability derivative with respect to any aerodynamic parameter. This has already been presented in Equations 9-23 and 9-24. The only components of the general F matrix that can be computed are those rows associated with supported degrees of freedom. Then the constraint:

$$\left[\frac{\partial C_F}{\partial \delta_{trim}}\right]_{lower} \le \frac{\partial C_F}{\partial \delta_{trim}} \le \left[\frac{\partial C_F}{\partial \delta_{trim}}\right]_{upper} \tag{4-30}$$

can easily be evaluated using the method of Table 11 where ε_{req} is the upper or lower bound as appropriate.

9.3.4. Trim Parameter Constraint

A constraint on any row of $\ddot{u}_r$ or δ may also be imposed using Equation 4-31. These data are obtained from the u_2 ($\equiv \ddot{u}_r$) and δ vectors that are recovered after the solution of Equation 9-22. Obviously, constraining a component of u_{2_k} or δ_s would result in an error since they are fixed with respect to design. Again, a constraint of the type shown in Table 8 is formed where ε_{req} is the user-specified bound.

9.4. ANTISYMMETRIC ANALYSES

Antisymmetric steady aerodynamic analyses are applied in ASTROS for the analysis and design of an aircraft's roll performance. For antisymmetric analyses, the u_l and δ vectors of Equation 9-19 have the following general forms:

$$\delta \in \left\{ \begin{array}{l} \mathbf{BETA} \\ \mathbf{PRATE} \\ \mathbf{RRATE} \\ \{\delta_{anti}\} \end{array} \right\} \qquad u_2 \in \left\{ \begin{array}{c} \mathbf{NY} \\ \mathbf{PACCEL} \\ \mathbf{RACCEL} \end{array} \right\}$$

9.4.1. Trim Analysis

The antisymmetric trim analysis follows exactly the symmetric trim analysis and recovery operations that are outlined in Section 9.3.1.

9.4.2. Aileron Effectiveness Constraint

Roll performance requirements frequently drive the design of aircraft wing structures. This factor has been recognized in ASTROS by the incorporation of an aileron effectiveness constraint. Aileron effectiveness, following terminology used in Reference 9 can be defined as the ratio of roll due to aileron deflection over roll due to roll rate:

$$\varepsilon_{eff} = -\frac{\left(C_{\ell_{\delta_a}}\right)^f}{\left(C_{\ell_{\frac{pb}{2V}}}\right)^f} \tag{4-28}$$

where Subsection 4.5.2.5 provides a definition of the terms used in this equation.

The effectiveness parameter is as a measure of the steady state roll rate achievable for a unit value of aileron deflection. In a manner similar to the lift effectiveness, the user can specify that the aileron effectiveness be within a specified range:

$$\varepsilon_{\min} \leq \varepsilon_{eff} \leq \varepsilon_{\max} \tag{9-41}$$

The stability derivatives required by Equation 4-28 can be determined using the right-hand side of Equation 9-19. The left-hand side of Equation 9-19 does not enter into this computation due to the specification that the effectiveness is computed for steady state roll (i.e., roll acceleration and therefore, $\boldsymbol{u}_2$, is zero). Explicitly, the right-hand side of Equations 9-19 is:

$$\frac{\bar{q}\,S\,b}{2}\,C_{\ell_{\delta_a}} = \left[\boldsymbol{P}_2 - \boldsymbol{K}_{21}\boldsymbol{K}_{11}^{-1}\boldsymbol{P}_1\right]\,\delta_{ail} = F_{NI}\,\delta_{ail} \tag{9-42}$$

$$\frac{\bar{q}\,S\,b^2}{4}\,C_{\ell_{\frac{pb}{2V}}} = \left[\boldsymbol{P}_2 - \boldsymbol{K}_{21}\boldsymbol{K}_{11}^{-1}\boldsymbol{P}_1\right]\,\delta_p = F_{NI}\,\delta_p \tag{9-43}$$

where δ_{ail} and δ_p are configuration vectors with unit values of aileron deflection and nondimensional roll rate, respectively, and F_{NI} is defined in Equation 9-31. The columns of rigid aerodynamic loads for the roll rate contained in $\boldsymbol{P}_2$ and $\boldsymbol{P}_1$ are computed for $\frac{p}{V}$ = 1.0. For this reason, an additional $\frac{b}{2}$ factor is required in the multiplication of the nondimensional stability derivative in Equation 9-43.

Given the stability derivatives of Equations 9-42 and 9-43, Equation 4-28 is used to determine the aileron effectiveness. The evaluation of the constraint is similar to that of Equation 9-40 and Table 11 with Equation 4-28 used for ε.

9.4.3. Flexible Stability Derivative Constraint

The formulation of 9.3.3 holds for antisymmetric analysis and is used without modification.

9.4.4. Trim Parameter Constraint

The formulation of 9.3.4 holds for antisymmetric analysis and is used without modification.

9.5. SENSITIVITY ANALYSIS

Calculation of gradient information for static aeroelasticity is quite similar to the derivation for static analysis sensitivities given in Subsection 6.3. This similarity is enhanced by the fact that the aerodynamic matrices of Equations 9-3 and 9-4 are invariant with respect to changes in the structural design. Finally, the δ vector sensitivity needs to be computed only as part of the trim analysis sensitivity and trim parameter constraint sensitivity calculations since δ is fixed for effectiveness calculations.

The meaning of a displacement set also varies, depending on the design condition. For the trim analysis, the displacements have the standard physical meaning of deformations induced by the specified flight condition. For the effectiveness constraints, these displacements give the deformation that would result from δ_α, δ_{ail}, and δ_p (see Equations 9-38, 9-42, and 9-43). There is also an acceleration vector that results from the unit angle of attack. These displacements and accelerations for the effectiveness constraints have minimal physical meaning, but their calculation is required to perform the sensitivity analysis.

With these remarks, the sensitivity of Equation 9-6 with respect to the i^{th} design variable in the *f-set* of displacements is

$$\boldsymbol{K}^{a}_{ff}\frac{\boldsymbol{u}_f}{\partial v_i} + \boldsymbol{M}_{ff}\frac{\ddot{\boldsymbol{u}}_f}{\partial v_i} = \boldsymbol{P}^{a}_{f}\frac{\partial \delta}{\partial v_i} + \frac{\partial \boldsymbol{R}_f}{\partial v_i} \tag{9-44}$$

where $\frac{\partial \boldsymbol{R}_f}{\partial v_i}$ has been defined following Equation 6-46.

For steady aerodynamic analyses, gravity and thermal loads are not allowed so that only the last two terms of Equation 6-37 contribute to $\frac{\partial \boldsymbol{R}_f}{\partial v_i}$ (i.e., only the stiffness and mass sensitivity values).

The orthogonality condition of Equation 9-12 requires a further sensitivity calculation. The sensitivity of this equation is:

$$-\begin{bmatrix}\boldsymbol{D} & \boldsymbol{I}\end{bmatrix}^T \begin{bmatrix}\boldsymbol{M}_{ll} & \boldsymbol{M}_{lr}\\ \boldsymbol{M}_{rl} & \boldsymbol{M}_{rr}\end{bmatrix} \begin{Bmatrix}\frac{\partial \boldsymbol{u}_l}{\partial v_i}\\ \frac{\partial \boldsymbol{u}_r}{\partial v_i}\end{Bmatrix} = -\begin{bmatrix}\boldsymbol{D} & \boldsymbol{I}\end{bmatrix}^T \begin{bmatrix}\frac{\partial \boldsymbol{M}_{ll}}{\partial v_i} & \frac{\partial \boldsymbol{M}_{lr}}{\partial v_i}\\ \frac{\partial \boldsymbol{M}_{rl}}{\partial v_i} & \frac{\partial \boldsymbol{M}_{rr}}{\partial v_i}\end{bmatrix} \begin{Bmatrix}\boldsymbol{u}_l\\ \boldsymbol{u}_r\end{Bmatrix} \tag{9-45}$$

where the fact that $\boldsymbol{D}$ is invariant with respect to the design variable is used. This equation is ultimately included as a constraint in the solution of the sensitivity of Equation 9-14. To assemble the necessary partitions, we start with:

$$\frac{\partial \boldsymbol{R}^{o}_{g}}{\partial v} \equiv \frac{\partial \boldsymbol{M}_{gg}}{\partial v_i}\boldsymbol{u}_g \tag{9-46}$$

The reduction of this vector to the *n-set* follows that given for applied loads:

$$\frac{\partial \boldsymbol{R}^{o}_{n}}{\partial v_i} = \frac{\partial \overline{\boldsymbol{R}}^{o}_{n}}{\partial v_i} + \boldsymbol{T}^{T}_{mn}\frac{\partial \boldsymbol{R}^{o}_{m}}{\partial v_i} \tag{9-47}$$

The single point constraints are removed by a partition of the *n-set* vectors to give $\frac{\partial \boldsymbol{R}^{o}_{f}}{\partial v_i}$ while the omitted degrees of freedom contribute to the *a-set*:

$$\frac{\partial \boldsymbol{R}^{o}_{a}}{\partial v_i} = \frac{\partial \overline{\boldsymbol{R}}^{o}_{a}}{\partial v_i} + \boldsymbol{G}_o\frac{\partial \boldsymbol{R}^{o}_{o}}{\partial v_i} \tag{9-48}$$

The pseudo-load vectors can be further partitoned into the *l-set* and *r-set* and an equation equivalent to that of Equation 9-12 can be written:

$$\begin{bmatrix} \boldsymbol{K}_{ll}^{a} & \boldsymbol{K}_{lr}^{a} & \boldsymbol{M}_{ll}\boldsymbol{D}+\boldsymbol{M}_{lr} \\ \boldsymbol{D}^{T}\boldsymbol{M}_{ll}+\boldsymbol{M}_{rl} & \boldsymbol{D}^{T}\boldsymbol{M}_{lr}+\boldsymbol{M}_{rr} & \boldsymbol{0} \\ \boldsymbol{D}^{T}\boldsymbol{K}_{ll}^{a}+\boldsymbol{K}_{rl}^{a} & \boldsymbol{D}^{T}\boldsymbol{K}_{lr}^{a}+\boldsymbol{K}_{rr}^{a} & \boldsymbol{m}_{r} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial \boldsymbol{u}_{l}}{\partial v_{i}} \\ \dfrac{\partial \boldsymbol{u}_{r}}{\partial v_{i}} \\ \dfrac{\partial \ddot{\boldsymbol{u}}_{r}}{\partial v_{i}} \end{Bmatrix} =$$

$$\begin{bmatrix} \boldsymbol{P}_{l}^{a} \\ \boldsymbol{0} \\ \boldsymbol{D}^{T}\boldsymbol{P}_{l}^{a}+\boldsymbol{P}_{r}^{a} \end{bmatrix} \frac{\partial \delta}{\partial v_{i}} + \begin{Bmatrix} \dfrac{\partial \boldsymbol{R}_{l}}{\partial v_{i}} \\ \boldsymbol{D}^{T}\dfrac{\partial \boldsymbol{R}_{l}^{o}}{\partial v_{i}}+\dfrac{\partial \boldsymbol{R}_{r}^{o}}{\partial v_{i}} \\ \boldsymbol{D}^{T}\dfrac{\partial \boldsymbol{R}_{l}}{\partial v_{i}}+\dfrac{\partial \boldsymbol{R}_{r}}{\partial v_{i}} \end{Bmatrix} \tag{9-49}$$

This equation can be rewritten, using the notation of 9-17a to become

$$\begin{bmatrix} \boldsymbol{K}_{11} & \boldsymbol{K}_{12} \\ \boldsymbol{K}_{21} & \boldsymbol{K}_{22} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial \boldsymbol{u}_{1}}{\partial v_{i}} \\ \dfrac{\partial \boldsymbol{u}_{2}}{\partial v_{i}} \end{Bmatrix}_{i} = \begin{Bmatrix} \boldsymbol{P}_{1} \\ \boldsymbol{P}_{2} \end{Bmatrix} \frac{\partial \delta}{\partial v_{i}} + \begin{Bmatrix} \dfrac{\partial \boldsymbol{R}_{1}}{\partial v_{i}} \\ \dfrac{\partial \boldsymbol{R}_{2}}{\partial v_{i}} \end{Bmatrix}_{i} \tag{9-50}$$

where $\dfrac{\partial \boldsymbol{u}_{l}}{\partial v}$ must be recovered from

$$\frac{\partial \boldsymbol{u}_{l}}{\partial v} = \boldsymbol{R}_{11}^{-1}\left[\boldsymbol{P}_{l}^{a}\frac{\partial \delta}{\partial v_{i}} + \frac{\partial \boldsymbol{R}_{l}}{\partial v} - \frac{\partial \boldsymbol{R}_{12}}{\partial v}\frac{\partial u_{r}}{\partial v} - \boldsymbol{R}_{13}\frac{\partial \ddot{u}_{r}}{\partial v}\right] \tag{9-51}$$

Equation 9-50 is the corollary to Equation 9-17a and is the basic equation for sensitivity analysis. Performing the evaluation of Equation 9-50 requires a rearrangement similar to that which occurs in Equations 9-20 to 9-22. The unknown values (either trim parameters, δ, or accelerations, $\boldsymbol{u}_2$ have sensitivity values while those that have been fixed do not. Using the notation of Equation 9-22, the basic evaluated trim sensitivities are

$$\left[\boldsymbol{K}_{22}-\boldsymbol{K}_{21}\boldsymbol{K}_{11}^{-1}\boldsymbol{K}_{12}\right]\frac{\partial \boldsymbol{u}_{2}}{\partial v_{i}} = \left[\boldsymbol{P}_{2}-\boldsymbol{K}_{21}\boldsymbol{K}_{11}^{-1}\boldsymbol{P}_{1}\right]\frac{\partial \delta}{\partial v_{i}}+\frac{\partial \boldsymbol{R}_{2}}{\partial v_{i}}-\boldsymbol{K}_{21}\boldsymbol{K}_{11}^{-1}\frac{\partial \boldsymbol{R}_{1}}{\partial v_{i}} \tag{9-52}$$

notice the $\boldsymbol{L}$ and $\boldsymbol{R}$ of Equation 9-20 appear here as well and need not be recomputed. Partitioning and rearranging the unknown sensitivities to the left hand side:

$$\begin{bmatrix} L_{ff} & R_{fu} \\ L_{kf} & R_{ku} \end{bmatrix} \begin{Bmatrix} \frac{\partial u_{2f}}{\partial v_i} \\ \frac{\partial \delta_u}{\partial v_i} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial P_{2f}}{\partial v_i} \\ \frac{\partial P_{2k}}{\partial v_i} \end{Bmatrix} - \begin{Bmatrix} \left[K_{21} K_{11}^{-1} \frac{\partial P_1}{\partial v_i} \right]_f \\ \left[K_{21} K_{11}^{-1} \frac{\partial P_1}{\partial v_i} \right]_k \end{Bmatrix} \tag{9-53}$$

where the zero derivative of the known fixed parameters has been exploited.

9.5.1. Trim Sensitivity Analysis

The basic trim sensitivity analysis involves the solution of Equation 9-53 for $\frac{\partial u_{2f}}{\partial v_i}$ and $\frac{\partial \delta_u}{\partial v_i}$. The vectors $\frac{\partial u_2}{\partial v_i}$ and $\frac{\partial \delta}{\partial v_i}$ are then obtained by merging the zero-valued sensitivities associated with the fixed parameters. For strength constraints, the sensitivities of the displacements must be recovered. The flexible displacement sensitivities are obtained using Equation 9-51, nothing that $u_1 \equiv u_r$ and $u_2 \equiv \ddot{u}_r$. The *f-set* sensitivities are computed by merging *o-set* and *a-set* sensitivities, where the *o-set* displacement sensitivities are obtained from:

$$\frac{\partial u_o}{\partial v_i} = G_o^a \frac{\partial u_a}{\partial v_i} + \left[K_{oo}^a \right]^{-1} P_o^a \frac{\partial \delta}{\partial v_i} \tag{9-54}$$

Further recovery to the *f-set* is a merge operation with the single point degrees of freedom. Equation 6-53 can then be used to complete the sensitivity analysis for strength and displacement constraints with steady aerodynmaics.

9.5.2. Lift Effectiveness Sensitivity

The calculation of the lift effectiveness sensitivity is most understandable if the third row of Equation 9-15 is used to compute the flexible lift curve slope:

$$\frac{\bar{q}S}{2} \frac{C_{L_{\alpha_f}}}{C_{m_{\alpha_f}}} = m_r \ddot{u}_r = \left[D^T P_l^a + P_r^a \right] \delta_\alpha - R_{31} u_{l_\alpha} - R_{32} u_{r_\alpha} \tag{9-55}$$

where $\begin{Bmatrix} u_{l_\alpha} \\ u_{r_\alpha} \end{Bmatrix}$ is the pseudo-deformation that results when the δ_α vector is applied to the free-free aircraft.

The sensitivity of Equation 9-51 gives:

$$\frac{\bar{q}S}{2}\begin{Bmatrix} C_{L_{\alpha_f}} \\ \hline \partial v \\ C_{m_{\alpha_f}} \\ \hline \partial v \end{Bmatrix}_i = -\begin{bmatrix} R_{31} & R_{32} \end{bmatrix}\begin{Bmatrix} \frac{\partial u_l^{\alpha}}{\partial v} \\ \frac{\partial u_r^{\alpha}}{\partial v} \end{Bmatrix}_i \tag{9-56}$$

Note that P_2, δ_α, R_{31} and R_{32} are invariant with respect to the design variable. The R_{31} and R_{32} matrices are invariant because the stiffness terms that are contained in this matrix (as defined by Equation 9-14) sum to zero, leaving only the design independent aerodynamic terms. To determine $\frac{\partial u_a^{\delta}}{\partial v}$, it is necessary to revisit Equation 9-50 and set $\frac{\partial \delta}{\partial v}$ to zero to obtain

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}\begin{Bmatrix} \frac{\partial u_r^{\alpha}}{\partial v_i} \\ \frac{\partial u_r^{\alpha}}{\partial v_i} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial R_1^{\alpha}}{\partial v_i} \\ \frac{\partial R_2^{\alpha}}{\partial v_i} \end{Bmatrix} \tag{9-57}$$

The first row of this equation gives

$$\frac{\partial u_r^{\alpha}}{\partial v_i} = K_{11}^{-1}\left[\frac{\partial R_1^{\alpha}}{\partial v_i} - K_{12}\frac{\partial u_r^{\alpha}}{\partial v_i}\right] \tag{9-58}$$

and this is substituted into the second row of Equation 9-53 and rearranged to give:

$$\left[K_{22} - K_{21}K_{11}^{-1}K_{12}\right]\frac{\partial \ddot{u}_r^{\alpha}}{\partial v_i} = \frac{\partial R_2^{\alpha}}{\partial v_i} - K_{21}K_{11}^{-1}\frac{\partial R_1^{\alpha}}{\partial v_i} \tag{9-59}$$

After solving Equation 9-59 for $\frac{\partial \ddot{u}_r^{\alpha}}{\partial v_i}$, Equation 9-58 can be used to determine $\frac{\partial u_r^{\alpha}}{\partial v_i}$ and Equation 9-56 is used to solve for the sensitivity of the stability derivative, and Equation 9-51 can be used to recover $\frac{\partial u_l^{\alpha}}{\partial v_i}$.

9.5.3. Aileron Effectiveness Sensitivity

In a manner similar to the lift effectiveness constraints, a more understandable formulation of the aileron effectiveness sensitivity can be gained by rewriting Equations 9-42 and 9-43

$$\frac{\bar{q}\,S\,b}{2}\,C_{\ell_{\delta_a}} = \left[D^T P_l^a + P_r^a\right]\delta_{ail} - R_{31}\,u_l^{ail} - R_{32}\,u_r^{ail} \tag{9-60}$$

$$\frac{\bar{q}\,S\,b^2}{4}\,C_{\ell_{\frac{pb}{2V}}} = \left[\boldsymbol{D}^T\boldsymbol{P}_l^a + \boldsymbol{P}_r^a\right]\delta_p - \boldsymbol{R}_{31}\,\boldsymbol{u}_l^p - \boldsymbol{R}_{32}\,\boldsymbol{u}_r^p \tag{9-61}$$

where $\boldsymbol{u}_l^{ail}$, $\boldsymbol{u}_r^{ail}$, $\boldsymbol{u}_l^p$ and $\boldsymbol{u}_r^p$ are the pseudo-deformations that result when the unit control surface, δ_{ail}, and unit roll rate δ_p vectors are applied, respectively. The sensitivity calculation therefore, requires the calculation and recovery to the *g-set* of these additional displacement vectors. The sensitivity of these stability derivatives are simply:

$$\frac{\partial C_{\ell_{\delta a}}}{\partial v_i} = -\frac{2}{qSb}\left[\boldsymbol{R}_{31} \quad \boldsymbol{R}_{32}\right]\begin{Bmatrix}\dfrac{\partial \boldsymbol{u}_l^{ail}}{\partial v_i}\\ \dfrac{\partial \boldsymbol{u}_r^{ail}}{\partial v_i}\end{Bmatrix} \tag{9-62}$$

$$\frac{\partial\, C_{\ell_{\frac{pb}{2V}}}}{\partial v_i} = -\frac{4}{qSb}\left[\boldsymbol{R}_{31} \quad \boldsymbol{R}_{32}\right]\begin{Bmatrix}\dfrac{\partial \boldsymbol{u}_l^{p}}{\partial v_i}\\ \dfrac{\partial \boldsymbol{u}_r^{p}}{\partial v_i}\end{Bmatrix} \tag{9-63}$$

where the $\frac{\partial \boldsymbol{u}_a^{ail}}{\partial v_i}$ and $\frac{\partial \boldsymbol{u}_a^{p}}{\partial v_i}$ vectors are the sensitivity of the pseudo displacements to the design variables and are calculated from Equation 9-52 using:

$$\frac{\partial \boldsymbol{u}_r^{ail}}{\partial v_i} = \boldsymbol{K}_{11}^{-1}\,\frac{\partial \boldsymbol{R}_1^{ail}}{\partial v_i} \tag{9-64}$$

$$\frac{\partial \boldsymbol{u}_r^{p}}{\partial v_i} = \boldsymbol{K}_{11}^{-1}\,\frac{\partial \boldsymbol{R}_1^{p}}{\partial v_i} \tag{9-65}$$

where the $\frac{\partial \boldsymbol{R}_1^{ail}}{\partial v_i}$ and $\frac{\partial \boldsymbol{R}_1^{p}}{\partial v_i}$ vectors are obtained from the pseudo-load vectors based on the pseudo-displacements. The simplified form of Equation 9-54 results from the fact that the $\frac{\partial \ddot{\boldsymbol{u}}_r}{\partial v_i}$ and $\frac{\partial \delta}{\partial v_i}$ vectors are null for the steady-state roll case.

9.5.4. Flexible Stability Derivative Sensitivities

A generalization of the approach used for lift effectiveness allows the computation of the sensitivities of flexible stability derivatives based on the displacements and accelerations that result from the solution of

the trim equation for a unit value of the associated parameter with all other parameters set to zero. The pseudo-displacements are then used in the computation of $\frac{\partial \boldsymbol{R}^{param}}{\partial v_i}$, which is formed:

$$\frac{\partial \boldsymbol{R}^{param}}{\partial v_i} = -\frac{\partial \boldsymbol{K}_{gg}}{\partial v_i}\boldsymbol{u}_g^{param} - \frac{\partial \boldsymbol{M}_{gg}}{\partial v_i}\ddot{\boldsymbol{u}}_g^{param} \tag{9-66}$$

along with a similar modification of Equation 9-46:

$$\frac{\partial \boldsymbol{R}_g^{o}}{\partial v_i} = \frac{\partial \boldsymbol{M}_{gg}}{\partial v_i}\boldsymbol{u}_g^{param} \tag{9-67}$$

Then Equation 9-52 can be used, with $\frac{\partial \delta^{param}}{\partial v_i} \equiv 0$, to obtain the sensitivity of the resultant rigid body accelerations:

$$\left[\boldsymbol{K}_{22} - \boldsymbol{K}_{21}\boldsymbol{K}_{11}^{-1}\boldsymbol{K}_{12}\right]\frac{\partial \ddot{\boldsymbol{u}}_r^{param}}{\partial v_i} = \frac{\partial \boldsymbol{R}_2^{param}}{\partial v_i} - \boldsymbol{K}_{21}\boldsymbol{K}_{11}^{-1}\frac{\partial \boldsymbol{R}_1^{param}}{\partial v_i} \tag{9-68}$$

and subsequently, using the design invariance of $\boldsymbol{R}_{31}$ and $\boldsymbol{R}_{32}$, and Equations 9-58 and 9-23:

$$\frac{\partial \boldsymbol{F}}{\partial v_i} = -\left[\boldsymbol{R}_{31} \quad \boldsymbol{R}_{32}\right]\begin{Bmatrix} \dfrac{\partial \boldsymbol{u}_l^{param}}{\partial v_i} \\ \dfrac{\partial \boldsymbol{u}_r^{param}}{\partial v_i} \end{Bmatrix} \tag{9-69}$$

from which any flexible stability derivative sensitivity, symmetric or antisymmetric, may be computed.

9.5.5. Trim Parameter Sensitivity

The solution of Equations 9-53, followed by the merging of the computed sensitivities of the fixed parameters results in the evaluation of $\frac{\partial \ddot{\boldsymbol{u}}_r}{\partial v_i}$ and $\frac{\partial \delta}{\partial v_i}$ which form the core of the sensitivities of these constraints.

THIS PAGE INTENTIONALLY LEFT BLANK

10. FLUTTER ANALYSIS

Flutter analysis in ASTROS provides the capability to assess the aeroelastic stability characteristics of the designed structure and to correct any deficiencies in a systematic fashion. Both subsonic and supersonic analyses are available and, reflecting the multidisciplinary character of the procedure, the design task can be performed with any number of boundary conditions and flight conditions. In this way, all critical flutter conditions can be analyzed and designed for simultaneously. This section first describes the flutter analysis that has been implemented in ASTROS and then describes the unique specification of the flutter constraints and the algorithm implemented to evaluate this constraint and the corresponding sensitivity calculation.

10.1. THE P-K FLUTTER ANALYSIS

The flutter analysis capability was implemented by combining software resources from FASTOP (Reference 5) and NASTRAN (Reference 1). The p-k method of flutter analysis was implemented based on an equation of the form

$$\left[\left(\frac{V}{b}\right)^2 p^2 \boldsymbol{M}_{hh} + \frac{V}{b} p \boldsymbol{B}_{hh} + \boldsymbol{K}_{hh} - \frac{\rho V^2}{2}\left(\boldsymbol{Q}_{hh}^R + \frac{p}{k}\boldsymbol{Q}_{hh}^I\right)\right] \boldsymbol{q}_h = \boldsymbol{0} \qquad (10\text{-}1)$$

where

V	selected airspeed
b	reference semi-chord
p	$\equiv k\ (\gamma + i)$ complex response frequency and eigenvalue
$\boldsymbol{M}_{hh}$	generalized mass matrix
$\boldsymbol{B}_{hh}$	generalized damping matrix for frequency response (Equations 11-8 and 11-4)
$\boldsymbol{K}_{hh}$	generalized stiffness matrix
$\boldsymbol{Q}_{hh}$	$= \left[\boldsymbol{Q}^R + i\boldsymbol{Q}^I\right]$ generalized aerodynamic matrix
ρ	air density
k	reduced frequency
$\boldsymbol{q}_h$	eigenvector of modal coordinates
γ	damping factor
i	$\equiv \sqrt{-1}$

Equation 10-1 is similar to the equation used in Reference 5 with the exception of the $\frac{p}{k}$ multiplier on the out-of-phase portion of the aerodynamics. This change was made to allow the proper evaluation of the aircraft's response at low, damped frequencies such as those required to estimate the aircraft's short period frequency.

The generation of the modal mass and stiffness matrices is performed as part of the dynamic matrix assembly described in Subsection 11.1. Options provided in this assembly allow for the possibility of direct matrix input and extra point degrees of freedom that can contribute to the off-diagonal terms in these matrices, but they are typically zero. A more commonly used feature is the specification of structural damping, which makes the stiffness matrix complex. Finally, although not specifically indicated in Equation 10-1, ASTROS has retained the FASTOP capability to omit designated modes from the flutter analysis. This feature is particularly useful when modes do not participate in the aeroelastic response and only obfuscate the interpretation of the analysis.

The computation of the $\boldsymbol{Q}_{hh}$ matrix at a number of Mach number and reduced frequency values is given in Equation 8-14. For a given Mach number, this matrix is calculated at a series of reduced frequencies (k's). Equation 10-1 requires this matrix as a continuous function of k, since the determination of values of p and k which satisfy Equation 10-1 is the basis of the p-k method. There are two different approaches in ASTROS to the interpolation of $\boldsymbol{Q}_{hh}$. The first approach follows that used in NASTRAN and fits a cubic through all the hard point k values specified in the user input KLIST (or, by default, all k values for the associated Mach number). The second approach uses separate linear, quadratic or cubic polynomials through the closest two, three or four hard point k values *for each term in* $\boldsymbol{Q}_{hh}$ and then computes the value at the required k value.

The NASTRAN algorithms are enumerated below:

$$G_{ij} = \begin{cases} \mathbf{0} & \text{for } i = j = nhdpts + 1 \\ \mathbf{1} & \text{for } i = nhdpts + 1 \text{ or } j = nhdpts + 1 \\ |k_i - k_j|^3 + |k_i + k_j|^3 & \text{for } i \text{ or } j \le nhdpts \end{cases} \tag{10-2}$$

where $nhdpts$ is the number of hard points (i.e., points at which $\boldsymbol{Q}_{hh}$ has been calculated) and the k_i are the reduced frequency values for these points. A weighting vector $\boldsymbol{C}$ is then determined

$$\boldsymbol{C} = \boldsymbol{G}^{-1}\, \boldsymbol{p}_v \tag{10-3}$$

where

$$p_{vj} = \begin{cases} \mathbf{1} & \text{for } j = nhdpts + 1 \\ |k_{est} - k_j|^3 + |k_{est} + k_j|^3 & \text{for } j \le nhdpts \end{cases} \tag{10-4}$$

where k_{est} is the reduced frequency value to which the aerodynamics are to be interpolated. The generalized aerodynamic matrix is then computed using

$$A_{hh}\left(k_{est}\right) = \sum_{j=1}^{nhdpts} C_j \left[Q_{hh}^{R}\left(k_j\right) + \frac{i}{k_j} Q_{hh}^{I}\left(k_j\right)\right] \tag{10-5}$$

where $\frac{Q_{hh}^{I}}{k_j}$ is fit rather than Q_{hh}^{I} directly since the former quantity is a much smoother value of k and because it is needed in the formulation of Equation 10-1.

In contrast, the local curve fit approach solves for the coefficients C of the following equations:

$$\begin{bmatrix} 1 & k_1 & k_1^2 & \cdots & k_1^{N-1} \\ 1 & k_2 & k_2^2 & \cdots & k_2^{N-1} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 1 & k_N & k_N^2 & \cdots & k_N^{N-1} \end{bmatrix} \begin{Bmatrix} C_{ij1}^{R} \\ C_{ij2}^{R} \\ \cdot \\ \cdot \\ C_{ijN}^{R} \end{Bmatrix} = \begin{Bmatrix} Q_{ij1}^{R} \\ Q_{ij2}^{R} \\ \cdot \\ \cdot \\ Q_{ijN}^{R} \end{Bmatrix} \tag{10-6}$$

and

$$\begin{bmatrix} 1 & k_1 & k_1^2 & \cdots & k_1^{N-1} \\ 1 & k_2 & k_2^2 & \cdots & k_2^{N-1} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 1 & k_N & k_N^2 & \cdots & k_N^{N-1} \end{bmatrix} \begin{Bmatrix} C_{ij1}^{I} \\ C_{ij2}^{I} \\ \cdot \\ \cdot \\ C_{ijN}^{I} \end{Bmatrix} = \begin{Bmatrix} \frac{1}{k_1} Q_{ij1}^{I} \\ \frac{1}{k_2} Q_{ij2}^{I} \\ \cdot \\ \cdot \\ \frac{1}{k_N} Q_{ijN}^{I} \end{Bmatrix} \tag{10-7}$$

where the real and imaginary parts of Q_{hh} are independently approximated by N^{th} order polynomials. In ASTROS, the order N may be linear ($N=1$), quadratic ($N=2$) or cubic ($N=3$) and the set $\left\{k_1, k_2, \ldots k_N\right\}$ is chosen as the set of 2, 3, or 4 $\left\{k_{hdpts}\right\}$ values closest to k_{est}. As k_{est} moves, the fits are recomputed when necessary if the set $\left\{k_1, \ldots k_N\right\}$ changes. A linear fit is used to extrapolate above or below the set of hard-point k's. The solution of the Vandermonde equations (10-6 and 10-7) is obtained using the algorithm of Reference 36. The terms of the generalized aerodynamic matrix are then obtained as:

$$A_{ij}\left(k_{est}\right) = \sum_{l=1}^{N} C_{ijl}^{R}\, k_{est}^{l-1} + i \sum_{l=1}^{N} C_{ijl}\, k_{est}^{l-1} \tag{10-8}$$

The solution algorithm for Equation 10-1 follows the one used in FASTOP. Figure 21 presents a basic flow chart of this process which involves solving the equation for a series of user defined velocity

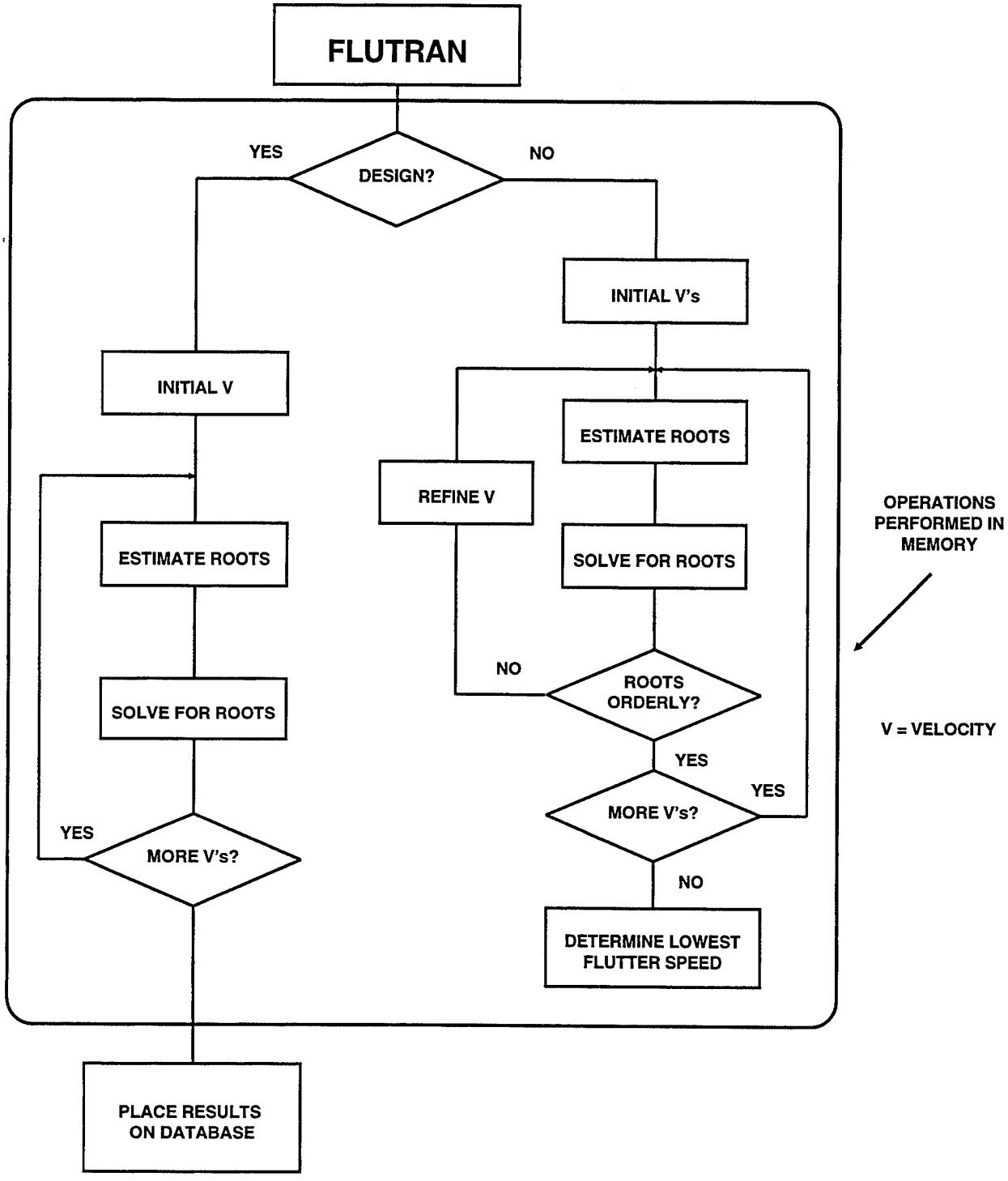


Figure 21. Flutter Analysis Algorithms Within ASTROS

values. The figure shows two alternative paths through the program, based on whether a flutter analysis or a flutter design task is being performed. The two differ in that the analysis refines the user-defined velocities to obtain a high quality display of the flutter response and, in particular, to determine the lowest flutter speed to a high degree of accuracy. As will be shown shortly, the design path does not require this refinement.

Equation 10-1 is solved by determining values of p for which the determinant of the equation is zero. FASTOP employs an algorithm based on Muller's method (Reference 19, pp 435-438). ASTROS adopted this algorithm, with the insertion of a capability to extract real roots that was not present in the Reference 5 software. The occurrence of real roots in the solution is not uncommon, and with the increasing use of active controls, is becoming more frequent. For real roots, the estimated damping is given by:

$$\gamma = \frac{p}{ln\,(2)} \tag{10-9}$$

10.2. FLUTTER CONSTRAINT EVALUATION

Flutter constraints are specified in ASTROS as

$$g = \frac{\gamma_{jl} - \gamma_{jREQ}}{\mathbf{GFACT}} \leq 0 \qquad \begin{matrix} j = 1, 2, \ldots, \text{nv} \\ l = 1, 2, \ldots, \text{nroot} \end{matrix} \tag{10-10}$$

where γ_{jl} is a damping value given by $\frac{\text{Re}\,(p)}{k}$ for the l^{th} root at the j^{th} velocity. γ_{jREQ} is the user defined, required damping value, with the j subscript indicating that the user can specify this requirement to be a function of velocity. Most typically, the required value would be zero for all velocities. **GFACT** is a scale factor that converts the damping numbers into a range consistent with other constraints in the design task. This is also a user input with suggested values in the range of 0.1 to 0.5.

The user specifies that this constraint be satisfied at a series of velocities up to, and perhaps above, the required flutter speed. Four or six velocities should be adequate. The advantages of this method of specifying the flutter analysis and constraint evaluation compared to various alternative methods are:

(1) There is no requirement for the computation of the flutter speed. The exact computation of this speed can consume substantial resources.

(2) By using the p-k method of flutter analysis, solutions are obtained only at the velocities of interest.

(3) The constraint is evaluated at multiple velocities to handle the appearance of "hump" modes that could become critical at velocities well below the required flutter speed. Flutter analysis at speeds that are 0.5, 0.75, 0.9, 1.0, and 1.1 times the required speed should be adequate for proscribing this undesirable behavior.

(4) In a similar fashion, the simultaneous consideration of a number of branches in the flutter solution handles the complication of more than one branch becoming critical. Also, when a

number of modes are considered, there is no necessity for tracking a specific mode, with its attendant increase in logical complexity.

(5) There is no large penalty associated with the calculation of the nroot x nv constraints given by Equation 10-10. This is because only the critical γ_{jl} conditions require gradient information. Very few such constraints are active for a typical design iteration.

10.3. SENSITIVITY OF FLUTTER CONSTRAINTS

The derivative of the constraint given by Equation 10-10 with respect to a design variable is

$$\frac{\partial g}{\partial v_i} = \frac{1}{\mathbf{GFACT}} \frac{\partial \gamma_{jl}}{\partial v_i} \tag{10-11}$$

The gradient of γ_{jl}, in turn is, from the definition following Equation 10-1 is

$$\frac{\partial \gamma_{jl}}{\partial v_i} = \frac{1}{k} \left(\frac{\partial \, \mathrm{Re}\,(p)}{\partial v_i} \right) - \gamma_{jl} \left(\frac{\partial \, \mathrm{Im}\,(p)}{\partial v_i} \right) \tag{10-12}$$

The gradients of the eigenvalues are based on Equation 10-1 in the physical coordinates, which can be condensed to

$$\Phi_{gh}^T \, \boldsymbol{F}_{gg} \, \Phi_{gh} \, \boldsymbol{q}_h = 0 \tag{10-13}$$

with an adjoint relation

$$\boldsymbol{y}_h^T \, \Phi_{gh}^T \, \boldsymbol{F}_{gg} \, \Phi_{gh} = 0 \tag{10-14}$$

The subscripts are suppressed for clarity in the remaining formulation. The derivative of Equation 10-13 with respect to design variable v_i is

$$2 \, \Phi^T \boldsymbol{F} \frac{\partial \Phi}{\partial v} \boldsymbol{q} + \Phi^T \frac{\partial \boldsymbol{F}}{\partial v} \Phi \, \boldsymbol{q} + \Phi^T \boldsymbol{F} \, \Phi \, \frac{\partial \boldsymbol{q}}{\partial v} = 0 \tag{10-15}$$

This equation can be pre-multiplied by $\boldsymbol{y}^T$ to give

$$2 \boldsymbol{y}^T \, \Phi^T \boldsymbol{F} \frac{\partial \Phi}{\partial v} \boldsymbol{q} + \boldsymbol{y}^T \, \Phi^T \frac{\partial \boldsymbol{F}}{\partial v} \Phi \, \boldsymbol{q} + \boldsymbol{y}^T \, \Phi^T \boldsymbol{F} \Phi \, \frac{\partial \boldsymbol{q}}{\partial v} = 0 \tag{10-16}$$

The third term in Equation 10-16 is zero from Equation 10-14. The first term in equation 10-16 is not typically zero. ASTROS, however, does not include this term in the computation of the flutter eigenvalue derivative. Since the normal modes used in the reduction to Equation 10-1 are updated at each iteration, the approximation made by ignoring the first term is justified in light of the computational expense associated with computing the eigenvector derivatives. Expanding the second term and reverting to modal coordinates gives

$$\boldsymbol{y}^T \left[\left(\frac{V}{b}\right)^2 p^2 \frac{\partial \boldsymbol{M}}{\partial v_i} + \left(\frac{V}{b}\right)^2 2p\, \boldsymbol{M} \frac{\partial p}{\partial v_i} + \left(\frac{V}{b}\right) \boldsymbol{B} \frac{\partial p}{\partial v_i} + p \left(\frac{V}{b}\right) \frac{\partial \boldsymbol{B}}{\partial v_i} + \frac{\partial \boldsymbol{K}}{\partial v_i} \right] \tag{10-17}$$
$$- \frac{\rho V^2}{2} \left[\frac{\partial \boldsymbol{A}^{\boldsymbol{R}}}{\partial v_i} + p \frac{\partial \boldsymbol{A}^{\boldsymbol{I}}}{\partial v_i} + \frac{\partial p}{\partial v_i} \boldsymbol{A}^I \right] \boldsymbol{q} = 0$$

where $\boldsymbol{A}^R$ and $\boldsymbol{A}^I$ are the real and imaginary portions of Equation 10-5 or 10-8, respectively. The velocity is fixed during the gradient evaluation so that the term $\frac{\partial V}{\partial v_i}$ is zero.

Left- and right-hand flutter eigenvectors in the global displacement set can be expressed as

$$\boldsymbol{y}_g = \varphi_{gh}\, \boldsymbol{y}_h \tag{10-18}$$
$$\boldsymbol{q}_g = \varphi_{gh}\, \boldsymbol{q}_h$$

Equation 10-17 can then be written in the g-set and the relations for the mass and stiffness gradients given by Equations 6-39 and 6-40 can be used. The solution of 10-17 is straightforward in the sense that the only unknown is the $\frac{\partial p}{\partial v_i}$ term. This can be solved for by two simultaneous linear equations, with the real and imaginary parts of the derivative the two unknowns. The notation for this is rather complex however, and it is convenient to define intermediate expressions:

$$MR_i + iMI_i = \boldsymbol{y}_g^T \frac{\partial \boldsymbol{M}_{gg}}{\partial v_i} \boldsymbol{q}_g \tag{10-19}$$

$$BR_i + iBI_i = \boldsymbol{y}_g^T \frac{\partial \boldsymbol{B}_{gg}}{\partial v_i} \boldsymbol{q}_g \quad ; \quad \frac{\partial \boldsymbol{B}_{gg}}{\partial v_i} = \frac{g}{\omega_3} \left[\frac{\partial \boldsymbol{K}_{gg}}{\partial v_i} + \alpha\, v_i^{\alpha-1} \boldsymbol{DKVB}_i \right] \tag{10-20}$$

$$KR_i + iKI_i = \boldsymbol{y}_g^T \left[\frac{\partial \boldsymbol{K}_{gg}}{\partial v_i} + \alpha v_i^{\alpha-1} \boldsymbol{DKVB}_i \right] \boldsymbol{q}_g \tag{10-21}$$

$$GMR + iGMI = \boldsymbol{y}_g^T \boldsymbol{M}_{gg} \boldsymbol{q}_g \tag{10-22}$$

$$GBR + iGBI = \boldsymbol{y}_g^T \boldsymbol{B}_{gg} \boldsymbol{q}_g \tag{10-23}$$

$$AIR + iAII = \boldsymbol{y}_h^T \boldsymbol{A}_{hh}^I \boldsymbol{q}_h \tag{10-24}$$

$$PR + iPI = p \tag{10-25}$$

$$P2R + iP2I = \frac{p^2 v^2}{b^2} \tag{10-26}$$

$$PVR + iPVI = \frac{p\,v}{b} \tag{10-27}$$

Note that these terms are all complex scalars.

The aerodynamic matrix is a function of the design variables through the reduced frequency. That is

$$\frac{\partial \boldsymbol{A}}{\partial v_i} = \frac{\partial \boldsymbol{A}}{\partial k}\,\frac{\partial k}{\partial v_i} \tag{10-28}$$

then, continuing to define simplified notation:

$$DAIR + iDAII = \boldsymbol{y}_h^T\,\frac{\partial \boldsymbol{A}^I}{\partial k}\,\boldsymbol{q}_h \tag{10-29}$$

$$DARR + iDARI = \boldsymbol{y}_h^T\,\frac{\partial \boldsymbol{A}^R}{\partial k}\,\boldsymbol{q}_h \tag{10-30}$$

and noting that $k = Im(p)$, further define

$$DR_i + iDI_i = \frac{\partial p}{\partial v_i} = \frac{\partial (k\gamma)}{\partial v_i} + i\,\frac{\partial k}{\partial v_i} \tag{10-31}$$

If the NASTRAN method of Aerodynamic interpolation is used, the gradient of the aerodynamic matrices with respect to k is a straightforward application of chain rule differentiation of Equation 10-5, with only the $\boldsymbol{C}_i$ term variable. The $\boldsymbol{G}$ matrix of Equation 10-3 is also invariant so that it is only the p_v vector of Equation 10-4 that requires differentiation:

$$\frac{\partial p_{v_i}}{\partial k} = 3a\,(k_{est} - k_j)^2 + 3\,(k_{est} - k_j) \tag{10-32}$$

If the local polynomial curve fit is used, the derivative of the aerodynamics is

$$\frac{\partial A_{lm}}{\partial v_i} = \sum_{n=2}^{N} (n-1)\; C^R_{lm_n}\; k_{est}^{(n-2)} + \sum_{n=2}^{N} (n-1)\; C^I_{lm_n}\; k_{est}^{n-2} \tag{10-33}$$

To further ease notation, the i subscript is implied in the following, with the understanding that Equation 10-17 must be solved for each active flutter constraint with respect to each design variable. With all this, Equation 10-17 becomes:

$$\begin{bmatrix} DF_{11} & DF_{12} \\ DF_{21} & DF_{22} \end{bmatrix} \begin{Bmatrix} DR \\ DI \end{Bmatrix} = \begin{bmatrix} P2R \cdot MR - P2I \cdot MI + PVR \cdot BR - PVI \cdot BI + KR \\ P2R \cdot MI + P2I \cdot MR + PVR \cdot BI + PVI \cdot BR + KI \end{bmatrix} \tag{10-34}$$

where

$$DF_{11} = \bar{q}AIR - \frac{2V^2}{b^2}\ (PR \cdot GMR - PI \cdot GMI) - \frac{V}{b} GBR$$

$$DF_{21} = \bar{q}AII - \frac{2V^2}{b^2}\ (PR \cdot GMI + PI \cdot GMR) - \frac{V}{b} GBI$$

$$DF_{12} = \bar{q}(-AII + PR \cdot DAIR - PI \cdot DAII + DARR) \qquad (10\text{-}35)$$

$$+ \frac{2V^2}{b^2}\ (PR \cdot GMI + PI \cdot GMR) + \frac{V}{b} GBI$$

$$DF_{22} = \bar{q}(AIR + PR \cdot DAII + PI \cdot DAIR + DARI)$$

$$- \frac{2V^2}{b^2}\ (PR \cdot GMR - PI \cdot GMI) - \frac{V}{b} GBR$$

where q is the dynamic pressure and the relation $\frac{\partial k}{\partial v_i} = DI$, from Equation 10-31, has been used. Note that the right-hand side of Equation 10-34 is independent of the design variable, so that these terms need to be calculated only once for each active flutter constraint.

Once Equation 10-34 has been solved for DR and DI, the required constraint gradients are computed using Equations 10-11 and 10-12 so that

$$\frac{\partial \gamma_{j1}}{\partial v_i} = \frac{1}{k}\left(DR - \gamma_{jl}\ DI\right) \qquad (10\text{-}36)$$

THIS PAGE INTENTIONALLY LEFT BLANK

11. DYNAMIC ANALYSIS

Dynamic analysis in ASTROS refers to analyses where the applied loading is a function of time or frequency. This section describes the ASTROS capability to perform transient and frequency analyses, with gust analysis treated as a special case of the frequency analysis. The additional special case of an aircraft's response to a blast type of loading is described in Section 12. Unlike the analyses described in the preceding five sections, there is no provision for considering the results of the dynamic response analysis in the design phase of ASTROS. The dynamic analysis capability is provided primarily to permit the checking of the final designs using these further analyses and to provide a more complete analysis package for general applications. The methodology described in this section borrows heavily from that developed for NASTRAN (Reference 15).

The basic equation for transient analysis is given by

$$\boldsymbol{M}\ddot{\boldsymbol{u}}+\boldsymbol{B}\dot{\boldsymbol{u}}+\boldsymbol{K}\boldsymbol{u} = \boldsymbol{P}(t) \tag{11-1}$$

and for frequency analyses by

$$\left[-\omega^2 \boldsymbol{M}+i\,\omega \boldsymbol{B}+\boldsymbol{K}+\boldsymbol{Q}\right]\boldsymbol{u} = \boldsymbol{P}(\omega) \tag{11-2}$$

where $\boldsymbol{M}$, $\boldsymbol{B}$ and $\boldsymbol{K}$ are the mass, damping and stiffness matrices and $\boldsymbol{Q}$ is the aerodynamic matrix that is used in the flutter and gust analyses.

This section first discusses the generation of the matrices on the left-hand side of these equations and then the generation of the time or frequency dependent load vectors on the right-hand side. The methods of solution used for each of the options developed for ASTROS is then given.

11.1. DYNAMIC MATRIX ASSEMBLY

The dynamic disciplines in ASTROS: flutter, transient response and frequency response, require additional operations to assemble the mass, damping, stiffness properties of the dynamic system(s) under analysis. This is done to accommodate those properties of the dynamic system which cannot be modeled directly using structural elements. The ASTROS dynamic matrix assembly is patterned after that in NASTRAN (Section 3 and Subsection 4.3 of Reference 15) and supports extra points, user defined direct matrix input and transfer function matrix input as well as several damping options to model the dynamic characteristics of the system. ASTROS does not provide damping elements (like the NASTRAN CVISC or CDAMP), nor is the NASTRAN feature for element dependent structural damping available in ASTROS. In keeping with the multidisciplinary nature of this code, ASTROS has introduced the innovation of having the extra point definitions include an "extra point set identification" which is used in the bound-

ary condition definition. The damping definition is also boundary condition dependent in ASTROS. These features allow several different dynamic systems to be analyzed simultaneously.

Dynamic matrix assembly in ASTROS and NASTRAN has a large number of options and so becomes very complex. Rather than duplicate the extensive discussion of this topic contained in Reference 15, this document emphasizes those features that are unique to ASTROS or are different than those in NASTRAN.

The analyses of the dynamic response disciplines can be done (in general) using either a direct or a modal formulation, although ASTROS does not support a direct formulation of the flutter analysis. Using NASTRAN as a guide to define the forms of the dynamic matrices, two forms of the mass and damping matrices (a direct form and a modal form) and four forms of the stiffness matrix: the transient and frequency response forms are different for both direct and modal formulations are available. Any or all of these eight matrices may be computed within each boundary condition in ASTROS, depending only on the selected dynamic disciplines and discipline options. Flutter analysis and optimization in ASTROS makes use of the modal frequency response form of the matrices. These forms are shown in Equations 11-3 through 11-10.

Direct forms:

$$\boldsymbol{M}_{dd} = \boldsymbol{M}^1_{dd} + \boldsymbol{M}^2_{dd} \tag{11-3}$$

$$\boldsymbol{B}_{dd} = \boldsymbol{B}^2_{dd} + \frac{g}{\omega_3}\boldsymbol{K}^1_{dd} \tag{11-4}$$

$$\boldsymbol{K}^t_{dd} = \boldsymbol{K}^1_{dd} + \boldsymbol{K}^2_{dd} \tag{11-5}$$

$$\boldsymbol{K}^f_{dd} = (1 + ig)\,\boldsymbol{K}^1_{dd} + \boldsymbol{K}^2_{dd} \tag{11-6}$$

Modal forms:

$$\boldsymbol{M}_{hh} = \boldsymbol{m}_h + \varphi^T_{dh}\,\boldsymbol{M}^2_{dd}\,\varphi_{dh} \tag{11-7}$$

$$\boldsymbol{B}_{hh} = \boldsymbol{b}_h + \varphi^T_{dh}\,\boldsymbol{B}^2_{dd}\,\varphi_{dh} \tag{11-8}$$

$$\boldsymbol{K}^t_{hh} = \boldsymbol{k}_h + \varphi^T_{dh}\,\boldsymbol{K}^2_{dd}\,\varphi_{dh} \tag{11-9}$$

$$\boldsymbol{K}^f_{hh} = (1 + ig)\,\boldsymbol{k}_h + \varphi^T_{dh}\,\boldsymbol{K}^2_{dd}\,\varphi_{dh} \tag{11-10}$$

where the subscripts d and h denote direct (*d-set*) and modal (*h-set*) forms, respectively and the superscripts t and f denote transient and frequency forms, respectively. The superscript *1* is used to denote those terms derived from the assembly of the structural elements and *2* to denote those terms obtained from direct matrix input or from transfer function input. The terms "g" and "3" refer to the general structural damping and the radian frequency used to define equivalent viscous damping, respectively.

The m_h are the generalized mass terms augmented with zeros for extra point degrees of freedom and φ_{dh} is the matrix of eigenvectors from the real eigenanalysis expanded to include extra points. The b_h are the expanded generalized modal damping terms obtained from an optional modal damping table, $g(\omega_h)$, defined by the user:

$$b_h = g(\omega_h) \cdot \omega_h m_h \tag{11-11}$$

and k_h are the generalized stiffness terms from the real eigenanalysis. Note that the expressions for the direct damping matrix and both frequency response stiffness matrices (Equations 11-4, 11-6 and 11-10) include both a complex structural damping and the viscous damping $\frac{g}{\omega_3}$. These terms are, however, mutually exclusive damping forms. If ω_3 is nonzero, viscous damping is used as in Equation 11-4 while a zero value for ω_3 results in the complex structural damping of Equations 11-6 and/or 11-10. More details on the ASTROS damping options are given in Subsection 11.1.3.

11.1.1. Direct Matrix Input

Direct matrix input allows the user to modify any or all of the dynamic mass, damping and stiffness matrices. ASTROS provides two mechanisms for the user to define direct matrix input. The most general is the direct matrix input option in which the user directly defines the matrices $\boldsymbol{M}_{dd}^2$, $\boldsymbol{B}_{dd}^2$, and/or $\boldsymbol{K}_{dd}^2$ in Equations 11-3 through 11-10. The second is through the definition of transfer functions. When these two methods are used in the same boundary condition, the resultant direct matrix input will be formed from the superposition of both sets of input.

Both direct matrix input and transfer functions refer to the physical or p-set degrees of freedom (where the p-set is the union of the structural degrees of freedom and the extra point degrees of freedom). In fact, the direct matrix input selected for dynamic matrix assembly must be square and of the order of the number of p-set degrees of freedom. The user can, therefore, couple structural degrees of freedom with the extra point degrees of freedom through both input mechanisms. The ASTROS feature for extra point sets adds a complication in that the size of the physical set varies between boundary conditions.

11.1.2. Reduction of Direct Matrix Input

The direct matrix input from all sources, which is formed in the p-set, is reduced to the dynamic set prior to its inclusion in the assembly process indicated in Equations 11-3 through 11-10. The extra point degrees of freedom create a complication in that the standard reduction matrices $\boldsymbol{T}_{mn}$ and $\boldsymbol{G}_o$ do not include extra points. In addition, matrix $\boldsymbol{G}_o$ may represent the result from either static condensation or from generalized dynamic reduction. The presence of extra points also requires that additional columns and rows be appended to the matrix of eigenvectors φ_{ai} that is output from the real eigenanalysis. These operations are:

$$\boldsymbol{T}_{mn}^d = \left[\boldsymbol{T}_{mn} \mid \boldsymbol{0} \right] \tag{11-12}$$

$$\left[G_o^d \right]_{static} = \left[G_o \mid 0 \right] \tag{11-13}$$

$$\left[G_o^d \right]_{gdr} = \begin{bmatrix} G_o & 0 \\ 0 & I \end{bmatrix} \tag{11-14}$$

$$\varphi_{dh} = \begin{bmatrix} \varphi_{ai} & 0 \\ 0 & I \end{bmatrix} \tag{11-15}$$

in which the *d* superscript or subscript denotes that the transformation applies to the reduction to the dynamic degrees of freedom (*d-set*). ASTROS has imposed the restriction that the extra point degrees of freedom follow all the structural degrees of freedom in the sequence list, so these operations can be performed by simply appending the proper terms to the appropriate partitioning vectors and transformation matrices.

Following this expansion of the transformation matrices, the standard matrix reductions are applied to the direct input matrices with the extra point degrees of freedom carried along in the independent, free and analysis sets. The modal transformations are applied in a separate step to the d-set direct input matrices if the modal forms of the dynamic matrices are required.

11.1.3. Damping Options

The damping options that are available in dynamic matrix assembly are sufficiently numerous that they merit additional clarification. Three means of specifying damping terms are available in ASTROS:

- the definition of a direct input damping matrix ***B2PP***, a complex direct input stiffness matrix, ***K2PP*** and/or specification of first order transfer function terms
- the specification of a structural damping value, *g*, and/or a radian frequency for equivalent viscous damping, ω_3
- the specification of a modal damping table, $g\ (\omega)$

The second and third options are selected for each boundary condition through the solution control boundary condition **DAMPING** option. For options two and three, the **DAMPING** option refers to **VSDAMP** and **TABDMP1** bulk data entries, respectively. These damping options may ALL coexist in a single boundary condition.

To understand the damping matrices that result for combinations of damping options, it is useful to understand the steps involved in the assembly of the damping and stiffness matrices. The direct input damping matrix $\boldsymbol{B}_{dd}^{2}$ is formed first from the transfer function and ***B2PP*** data, if any are selected in the solution control boundary condition definition. The direct damping matrix $\boldsymbol{B}_{dd}$ is then assembled as shown in Equation 11-4 with the second term omitted unless both "*g*" and "ω_3" are defined through the **DAMPING** option. This assembly process does not depend on whether the modal or direct formulation is desired since the second term in Equation 11-8 is the modal transformation of the full direct damping

matrix. Thus, when equivalent structural damping is selected, it will appear in both the direct and modal damping matrices.

Once the direct damping matrix is formed, the assembly of the modal damping matrix may proceed. Unless a modal damping table is referenced by the **DAMPING** option, the modal damping matrix is merely the modal transformation of the direct damping matrix. If, however, the **DAMPING** option refers to a modal damping table (note that the **DAMPING** option can refer to both **VSDAMP** and a modal damping table in the same boundary condition), the b_h terms are formed and included in the assembly of the modal damping matrix.

For both direct and modal frequency response (and flutter analysis), the complex structural damping option is available. In this case, the complex multiplier is applied to the structural stiffness matrix or the generalized stiffness matrix as shown in Equations 11-6 and 11-10. The multiplier will be unity, however, if the equivalent viscous damping option has been selected instead. That is the case if both "g" and "ω_3" are nonzero on the referenced **VSDAMP** entry or if there is no **VSDAMP** entry selected. The frequency response forms of both the modal and direct stiffness matrices might not, therefore, be complex matrices if the imaginary term is zero. There is no restriction, however, that the direct matrix input of ***K2PP*** be real, so that complex structural damping input through direct matrix input can coexist with the equivalent structural damping of Equation 11-4. Therefore, the frequency response modal and/or direct stiffness matrices may be complex even though the equivalent structural damping option is selected.

11.2. DYNAMIC LOADS GENERATION

ASTROS has adapted NASTRAN loads generation concept to define the right-hand sides of Equations 11-1 and 11-2. The formats used in ASTROS for the preparation of the user input for these loads has been modified from the NASTRAN formats and can be quite involved. Subsection 3.5 of the Applications Manual provides guidance on this preparation. This subsection is limited to a specification of the types of loads input that are available for the dynamic response analyses.

11.2.1. Transient Loads

For transient loads, the $\boldsymbol{P}(t)$ vector of Equation 11-1 is specified as the weighted sum of any number of component loads:

$$\boldsymbol{P}(t) = S_o \sum_i S_i \boldsymbol{L}_i(t) \tag{11-16}$$

where S_o and S_i are scalar multipliers. Note that this is similar to static loads generation. The $L(t)$ vector, in turn, can be represented as the product of a spatial component and a time varying component

$$\boldsymbol{L}(t) = \boldsymbol{X}_t \cdot \boldsymbol{T} \tag{11-17}$$

where this matrix notation is meant to convey the information that any number of time functions can be specified for a given model. The $\boldsymbol{X}_t$ matrix has as many rows as there are degrees of freedom in the p-set and as many columns as there are unique time functions. The $\boldsymbol{T}$ matrix has as many rows as there are

unique time functions and a column for each time step that the user has requested. There are two distinct formats for specifying the rows of the $\boldsymbol{T}$ matrix. The first is a general form of

$$\boldsymbol{T}_{ij} = F_i(t_j - \tau_i) \tag{11-18}$$

where τ_i is a user input and the F_i functions are input as a tabular function of time. The second format is the specialized form of

$$\boldsymbol{T}_{ij} = \begin{cases} \bar{t}_j^{b_i}\, e^{c_i \bar{t}_j} \cos(\omega_i \bar{t}_j + \varphi_i) & 0 \le \bar{t}_j \le T_{2_i} - T_{1_i} \\ 0 & 0 \le \bar{t}_j \text{ and } \bar{t}_j > T_{2_i} - T_{1_i} \end{cases} \tag{11-19}$$

where $\bar{t}_j = t_j - T_{1_i} - \tau_i$, τ_i, b_i, c_i, ω_i, φ_i, T_{1_i} and T_{2_i} are user inputs and the F_i functions of Equation 11-18 are input as a tabular function of time. The actual input of the special functions using Equation 11-19 is perhaps easier in practice than it is in theory. This is because most of the input terms are likely to be zero for particular wave forms.

ASTROS generates the $\boldsymbol{X}_t$ matrix of Equation 11-17 in the p-set. Before the multiplication by the T matrix is performed, the $\boldsymbol{X}_t$ matrix is reduced to d- or h-size, depending on whether a direct or modal formulation has been specified. The scalar multiplications of Equation 11-16 are also performed on the spatial matrices prior to multiplication by the $\boldsymbol{T}$ matrix. Following the multiplication, the loads are stored on the data base for later retrieval in the response calculation. The $\boldsymbol{P}$(t) matrix has dimensions of either d-size by **NSTEP**, the number of time steps in the response, or of h-size by **NSTEP**.

11.2.2. Frequency Dependent Loads

In a manner similar to the transient loads, the frequency dependent loads of Equation 11-2 are generated as the weighted sum of any number of component loads:

$$\boldsymbol{P}(\omega) = S_o \sum_i S_i \boldsymbol{L}_i(\omega) \tag{11-20}$$

where S_o and S_i are scalar multipliers. The L (ω) matrix is, in turn, represented as the product of a spatial component and a frequency dependent component:

$$\boldsymbol{L}(\omega) = \boldsymbol{X}_f \cdot \boldsymbol{F} \tag{11-21}$$

where this matrix notation is meant to convey the information that any number of frequency functions can be specified for a given model. The $\boldsymbol{X}_f$ matrix has as many rows as there are degrees of freedom in the p-set and as many columns as there are unique frequency functions. The $\boldsymbol{F}$ matrix has as many rows as there are unique frequency functions and **NFREQ**, the number of frequencies required for the response

analysis, columns. As in the transient load case, there are two formats for specifying elements in the ***FQ*** matrix. The first is

$$F_{ij} = \left[C_i(f_j) + i D_i(f_j) \right] e^{i\left(\theta_i - 2\pi f_j \tau_i\right)} \tag{11-22}$$

while the second is

$$F_{ij} = \left[B_i(f_j)\ e^{i\theta_i\left(f_j\right)} \right] e^{i\left(\varphi_i(f_j) - 2\pi f_j \tau_i\right)} \tag{11-23}$$

where B_i, $\varphi_i(f_j)$, C_i and D_i are input as tabular functions of frequency and θ_i and τ_i are user inputs. The f_j values are the user specified frequencies at which the response is to be calculated.

ASTROS generates the X_f matrix of Equation 11-21 in the *p-set*. Before the multiplication by the ***F*** matrix is performed, this matrix is reduced to *d-size* or *h-size*, depending on whether a direct or modal formulation has been specified. The scalar multiplications of Equation 11-20 are also performed prior to the multiplication by the ***F*** matrix. Following the multiplication, the loads are stored on the data base for later retrieval in the response calculation. The $P(\omega)$ matrix has dimensions of either d-size by **NFREQ** or h-size by **NFREQ**.

11.2.3. Gust Loads

Gust analysis in ASTROS is performed as a special type of frequency analysis. As discussed in Subsection 8.2.2, if a gust analysis is being performed, the Q_{hh} matrix of Equation 8-14 is computed to provide forces due to aeroelastic deformations while the Q_{hj} matrix of Equation 8-15 is computed to provide the gust loads on the rigid aircraft.

The overall gust load is computed by combining these Q_{hj} data with a downwash vector and a frequency dependent shaping function, as described in the following paragraphs.

A one-dimensional sinusoidal gust field produces a downwash vector, ***Wj***, at the aerodynamic panels that has elements of the form

$$Wj(\omega) = \cos\gamma_j \cdot \exp\left[\frac{-i\omega(x_j - x_o)}{V} \right] \tag{11-24}$$

where

ω = Frequency

j = Panel number

γ = Panel dihedral angle

$x_j - x_o$ = Distance from the user input reference plane to the aerodynamic panel

V = Vehicle velocity

The downwash vector can be thought of as a mode shape which can be multiplied by the $\boldsymbol{Q}_{hj}$ aerodynamic operator to give unit gust loads in modal coordinates:

$$\boldsymbol{P}_{\delta}(\omega) = \boldsymbol{Q}_{hj}(\omega)\ \boldsymbol{Wj}(\omega) \tag{11-25}$$

As in the flutter analysis, the $\boldsymbol{Q}_{hj}$ matrix is required at a number of frequencies while it typically has been computed at a different, smaller set of frequencies. The interpolation scheme described in Equations 10-2 through 10-5 for the $\boldsymbol{Q}_{hh}$ matrix is applied to the $\boldsymbol{Q}_{hj}$ matrix as well. The interpolation scheme of Equation 10-8 is not available for gust analysis.

As a final step in gust load generation, the ***PDEL*** matrix can be modified by a user defined function of frequency and by a gust velocity scale factor:

$$\boldsymbol{P}_{h}^{f}(\omega) = \bar{g}\, wg\ \boldsymbol{P}_{w}(\omega)\ \boldsymbol{P}_{\delta}(\omega) \tag{11-26}$$

where

$\boldsymbol{P}_{h}^{f}(\omega)$ = Gust load vector in modal coordinates and is equivalent to the matrix of Equation 11-19
$\bar{q}$ = Dynamic pressure
w_g = Gust scale factor
$\boldsymbol{P}_{w}(\omega)$ = A frequency dependent weighting function matrix

where the $\boldsymbol{P}_{w}(\omega)$ function is defined using one of the tabular forms specified by Equations 11-22 or 11-23.

11.3. TRANSIENT RESPONSE ANALYSIS

As described in the preceding subsection, Equation 11-1 can be specified in terms of modal or direct coordinates. If a modal analysis is specified, ASTROS checks whether the equations are coupled or uncoupled. This is done by checking if the M, B and K matrices are all diagonal. If they are, then the equations are solved in a relatively efficient manner using analytical equations. If they are coupled, the Newmark-Beta numerical technique is employed. Each of these methods is now discussed using terminology given in Subsection 4.6 of Reference 15. The further option of using Fast Fourier Transform techniques to perform the transient analysis is also described.

11.3.1. Solution of Uncoupled Transient Response Equations

If the modal equations are uncoupled, it is possible to write each row of Equation 11-1 separately:

$$m_i\ \ddot{q}_i + b_i\ \dot{q}_i + k_i\ q_i = P_i(t) \tag{11-27}$$

which can be put into a more standard form as:

$$\ddot{q}_i + 2\beta\ \dot{q}_i + \omega_o^2\ q_i = \frac{P_i(t)}{m_i} \tag{11-28}$$

where

$$\beta = \frac{b_i}{2m_i}$$

$$\omega_o^2 = \frac{k_i}{m_i}$$

Equation 11-28 can be solved for the response at any time in terms of the displacement and velocity at specified times t_n and a convolution integral of the applied load:

$$q_i(t) = F(t-t_n)\ q_{i,n} + G(t-t_n)\ \dot{q}_{i,n} + \frac{1}{m_i}\int_{t_n}^{t} G(t-\tau)\ P_i(\tau)\, d\tau \tag{11-29}$$

where the F and G functions are combinations of the homogeneous solutions

$$q_i(t) = \exp\left[\left(-\beta \pm \sqrt{\beta^2 - \omega_o^2}\right)\left(t-t_n\right)\right] \tag{11-30}$$

F and G satisfy, respectively, the initial conditions for unit displacement and unit velocity.

It is assumed that the load varies linearly between t_n and $n+1$, so that, in Equation 11-29

$$P_i(\tau) = P_{i,n} + \frac{\tau}{\Delta t}\left(P_{i,n+1} - P_{i,n}\right) \tag{11-31}$$

For this form of the applied load, the integral in Equation 11-29 can be evaluated in closed form. The general form of the solutions at the next time step, $t = t_{n+1}$, in terms of the initial conditions at $t = t_n$ and the applied loads, is

$$q_{i,n+1} = F\ q_{i,n} + G\ q_{i,n} + A\ P_{i,n} + B\ P_{i,n+1} \tag{11-32}$$

$$\dot{q}_{i,n+1} = F'\ q_{i,n} + G'\ q_{i,n} + A'\ P_{i,n} + B'\ P_{i,n+1} \tag{11-33}$$

The coefficients are functions of the modal parameters, m_i, β, ω_o^2, and of the time increment, Δt. The uncoupled modal solutions are evaluated at all time steps by recurrent application of Equations 11-32 and 11-33. The accelerations are calculated by solving for $\ddot{q}$ from Equation 11-28:

$$\ddot{q}_{i,n+1} = \frac{P_{i,n+1}}{m_i} = 2\beta q_{i,n+1} - \omega_o^2 q_{i,n+1} \tag{11-34}$$

The algebraic expressions for the coefficients in Equations 11-32 and 11-33 depend on whether the homogeneous solutions are under damped $\omega_o^2 > \beta^2$, critically damped $\omega_o^2 = \beta^2$, or over damped $\omega_o^2 < \beta^2$. In addition, a separate set of expressions is used for undamped rigid body modes $\omega_o = \beta_o = 0$. As an example, these terms are defined here for the most frequently encountered case of underdamped solutions while Subsection 11.4 of Reference 1 provides definition for all four cases.

$$F = e^{-\beta \Delta t}\left(\cos \omega \Delta t + \frac{\beta}{\omega}\sin \omega \Delta t\right)$$

$$G = \frac{1}{\omega} e^{-\beta \Delta t} \sin \omega \Delta t$$

$$A = \frac{1}{\Delta t\, k\, \omega}\left[e^{-\beta \Delta t}\left[\left(\frac{\omega^2 - \beta^2}{w_o^2} - \Delta t\, \beta\right)\sin \omega \Delta t - \left(\frac{2\omega\beta}{w_o^2} + \Delta t\, \omega\right)\cos \omega \Delta t\right] + \frac{2\beta\omega}{w_o^2}\right]$$

$$B = \frac{1}{\Delta t\, k\, \omega}\left[e^{-\beta \Delta t}\left[-\left(\frac{\omega^2 - \beta^2}{w_o^2}\right)\sin \omega \Delta t + \frac{2\omega\beta}{w_o^2}\cos \omega \Delta t\right] + \omega \Delta t\, \frac{2\beta\omega}{w_o^2}\right] \tag{11-35}$$

$$F' = -\frac{\omega_o^2}{\omega} e^{-\beta \Delta t} \sin \omega \Delta t$$

$$G' = e^{-\beta \Delta t}\left(\cos \omega \Delta t - \frac{\beta}{\omega}\sin \omega \Delta t\right)$$

$$A' = \frac{1}{\Delta t\, k\, \omega}\left[e^{-\beta \Delta t}\left[(\beta + \Delta t\, \omega_o^2)\sin \omega \Delta t + \omega \cos \omega \Delta t\right] - \omega\right]$$

$$B' = \frac{1}{\Delta t\, k\, \omega}\left[- e^{-\beta \Delta t}(\beta \sin \omega \Delta t + \omega \cos \omega \Delta t) + \omega\right]$$

where

$$\omega^2 = \omega_o^2 - \beta^2 \quad \text{and} \quad k = \omega_o^2 m_i$$

11.3.2. Solution of Coupled Transient Response Coupled Equations

If the modal equations contain off-diagonal terms or if the direct method of analysis is used, the uncoupled formulation of the preceding subsection is not applicable. Instead a numerical procedure must

be adopted and ASTROS has selected the Newmark-Beta algorithm used in NASTRAN. This method transforms Equation 11-1 to a discrete equivalent of the form

$$\boldsymbol{A}\ \boldsymbol{u}_{n+1} = \frac{1}{3}\left[\boldsymbol{P}_{n+1} + \boldsymbol{P}_n + \boldsymbol{P}_{n-1}\right] + \boldsymbol{C}\ \boldsymbol{u}_n + \boldsymbol{D}\ \boldsymbol{u}_{n-1} \tag{11-36}$$

where

$\boldsymbol{u}_i$ = displacement response at the i^{th} time step

$$\boldsymbol{A} = \frac{1}{\Delta t^2}\boldsymbol{M} + \frac{1}{2\Delta t}\boldsymbol{B} + \frac{1}{3}\boldsymbol{K}$$

$$\boldsymbol{C} = \frac{2}{\Delta t^2}\boldsymbol{M} - \frac{1}{3}\boldsymbol{K}$$

$$\boldsymbol{D} = -\frac{1}{\Delta t^2}\boldsymbol{M} + \frac{1}{2\Delta t}\boldsymbol{B} - \frac{1}{3}\boldsymbol{K}$$

Δt = time step

For a fixed time step, matrices $\boldsymbol{A}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ need to be computed only once. Additionally, $\boldsymbol{A}$ is decomposed so that the loop on the time step only requires forward/backward substitutions to solve Equation 11-36.

Equation 11-36 requires the response and the load at two previous steps, as well as the load at the current time step. In order to initiate the calculation, starting values are calculated using

$$\begin{aligned}
\boldsymbol{u}_{-1} &= \boldsymbol{u}_o - \dot{\boldsymbol{u}}_o\ \Delta t \\
\boldsymbol{P}_{-1} &= \boldsymbol{K}\ \boldsymbol{u}_{-1} + \boldsymbol{B}\ \dot{\boldsymbol{u}}_o \\
\boldsymbol{P}_o &= \frac{1}{2}\left[\overline{\boldsymbol{P}}_o + \boldsymbol{K}\ \boldsymbol{u}_o + \boldsymbol{B}\ \dot{\boldsymbol{u}}_o\right]
\end{aligned} \tag{11-37}$$

where $\overline{\boldsymbol{P}}_o$ is the user input load vector at the initial time. Initial conditions (i.e., $\boldsymbol{u}_o$ and $\dot{\boldsymbol{u}}_o$) are available only for the direct method. When the time step changes, the matrices of Equation 11-36 need to be recomputed. Also, the starting values need to be adjusted to:

$$\begin{aligned}
\boldsymbol{u}_{-1} &= \boldsymbol{u}_n - \Delta t_2\ \dot{\boldsymbol{u}}_o + \frac{1}{2}\Delta t_2^2\ \ddot{\boldsymbol{u}}_o \\
\boldsymbol{P}_{-1} &= \boldsymbol{K}\ \boldsymbol{u}_{-1} + \boldsymbol{B}\ \dot{\boldsymbol{u}}_o - \Delta t_2\ \boldsymbol{u}_o + \boldsymbol{M}\ \ddot{\boldsymbol{u}}_o
\end{aligned} \tag{11-38}$$

where

Δt_2 = new time step

$$\dot{u}_o = \frac{1}{\Delta t_1}\left[u_n - u_{n-1}\right]$$

$$\ddot{u}_o = \frac{1}{\Delta t_1^2}\left[u_n - 2u_{n-1} + u_{n-2}\right]$$

n = last time of the old time step

Δt_1 = old time step

Equation 11-35 provides displacement information. If velocity and acceleration information are also required, these vectors are calculated using

$$\dot{u}_n = \frac{1}{2\Delta t}\left[u_{n+1} - u_{n-1}\right] \tag{11-39}$$

$$\ddot{u}_n = \frac{1}{\Delta t^2}\left[u_{n+1} - 2u_n + u_{n-1}\right] \tag{11-40}$$

11.3.3. Solution of Transient Equations Using Fast Fourier Transforms

A third transient analysis technique transforms the time dependent loads into the frequency domain and using Fast Fourier Transform (FFT) techniques, solves the equations using the frequency response techniques of Subsection 11.4 and then transforms the resulting response functions back to the time domain. Appendix C contains a description of the FFT algorithms.

If the FFT option is specified, the time dependent loads of the $\boldsymbol{T}$ matrix in Equation 11-18 or 11-19 are computed as equal time intervals as specified by the user. Each row of the $\boldsymbol{T}$ matrix is transformed independently. No restrictions are imposed by ASTROS on the form of this time function, but it must conform to the restrictions of periodicity or be of sufficiently short duration that FFT methods are applicable.

Once the frequency response has been calculated, the inverse FFT algorithm is applied separately to each degree of freedom in the response. This provides the response of the displacement. The response of the velocity is obtained by multiplying the frequency domain data by $i\,\omega$, the imaginary constant times each frequency value, and performing the inverse FFT on the resulting frequency vector. Similarly, the acceleration response is obtained by performing the inverse FFT on the displacement response in the frequency domain multiplied by $-\omega^2$.

11.4. FREQUENCY RESPONSE ANALYSIS

As in the transient response case, the frequency response calculation of Equation 11-2 can be performed in terms of direct or modal coordinates. If a modal analysis is specified, a determination is made whether the equations are coupled or uncoupled by checking if the $\boldsymbol{M}$, $\boldsymbol{B}$ or $\boldsymbol{K}$ matrices are all diagonal. If they are, then the equations can be solved independently in a relatively efficient fashion. The

following subsections described the response calculations for both the uncoupled and the coupled formulations. The special case of performing gust analysis in the frequency domain is treated in a separate subsection.

11.4.1. Solution of Uncoupled Frequency Response Equations

If the modal equations are uncoupled, it is possible to write each row of Equation 11-2 separately:

$$\left(-\omega^2 \, m_i + i\omega b_i + k_i \right) q_i = P_i(\omega) \tag{11-41}$$

Equation 11-41 is solved for each frequency and mode combination to give the overall frequency response.

11.4.2. Solution of the Coupled Frequency Response Equations

If the matrices given in Equation 11-2 are coupled, the response is calculated using

$$\boldsymbol{u} = \left[-\omega^2 \boldsymbol{M} + i\omega \boldsymbol{B} + \boldsymbol{K} \right]^{-1} \boldsymbol{P}(\omega) \tag{11-42}$$

This indicates that a separate decomposition of the matrix must be performed for each frequency in the analysis. ASTROS uses standard decomposition and forward/backward substitution routines to solve Equation 11-42.

11.4.3. Solution of Frequency Response Equation Including Gusts

If gust loads are present for the analysis, the solution technique of Equation 11-42 can still be applied, but it is necessary to add terms representing the aerodynamic effects. The direct solution option is not supported for this case so that the equation to be solved is

$$\left[-\omega^2 \boldsymbol{M}_{hh} + i\,\omega \left[\boldsymbol{B}_{hh} - \frac{qb}{v} \boldsymbol{A}^I_{hh} \right] + \boldsymbol{K}_{hh} - \boldsymbol{A}^R_{hh} \right] \boldsymbol{u}_h = \boldsymbol{P}^f_h(\omega) \tag{11-43}$$

where the $\boldsymbol{A}^R_{hh}$ and $\boldsymbol{A}^I_{hh}$ matrices are the real and imaginary parts of Equation 10-5 and the $\boldsymbol{P}^f_h$ vector is defined in Equation 11-26.

THIS PAGE INTENTIONALLY LEFT BLANK

12. NUCLEAR BLAST RESPONSE

12.1. INTRODUCTION

A further dynamic analysis incorporated into ASTROS is the calculation of the response of an aircraft to an encounter with a blast created by a nuclear explosion. Although this represents a very specialized analysis, the nuclear blast response calculation has a high degree of commonality with the core ASTROS disciplines; i.e., results from the structural analysis, unsteady aerodynamics and transient analysis calculations already discussed in the previous sections all provide part of the information required for this integrated analysis. The implementation of this capability into ASTROS was affected through a combination of Northrop supplying the basic ASTROS system and integration tasks while Kaman AviDyne, in a subcontractor capacity, supplied the code that performs the specialized calculations related to nuclear blast encounters. This section presents an overview of the blast response calculations while Appendix B contains a description of the details of the fitting process which converts frequency dependent aerodynamics into the time domain that was prepared by Kaman AviDyne.

Figure 22 presents a block diagram of the general form of the blast response calculations. As the figure indicates, the overall calculations are divided into two distinct sections. The first is a preprocessing section shown in the lower portion of the figure which converts frequency dependent aerodynamic influence coefficient matrices into a number of indicial functions from which the gust load can be computed as a function of time. The lower portion contains the capability to convert forces acting on individual boxes to generalized forces acting on structural modes. The upper portion depicts the actual calculation of the blast response. There is a feedback loop in this calculation in that the total forces acting on the structural modes are a function not only of the blast wave and the aircraft position, but also of the structural modes themselves. The following two subsections briefly describe how each of these portions was integrated into the ASTROS system.

12.2. THE AERODYNAMICS PREPROCESSOR

The calculation of the aircraft response during an encounter with a nuclear blast is computed in ASTROS in the time domain. This is in contrast to the related atmospheric gust response calculation of Subsection 11.4.3 which is performed in the frequency domain. The selection of the time domain was partially done for historical reasons, since the VIBRA series of computer programs that have been developed by Kaman AviDyne (Reference 27) all compute the response as a function of time. More importantly, nonlinear effects can be readily accounted for once the equations are in the time domain. While the current implementation of the blast response calculations is linear, it does serve as a basis for more complex formulations if they become necessary.

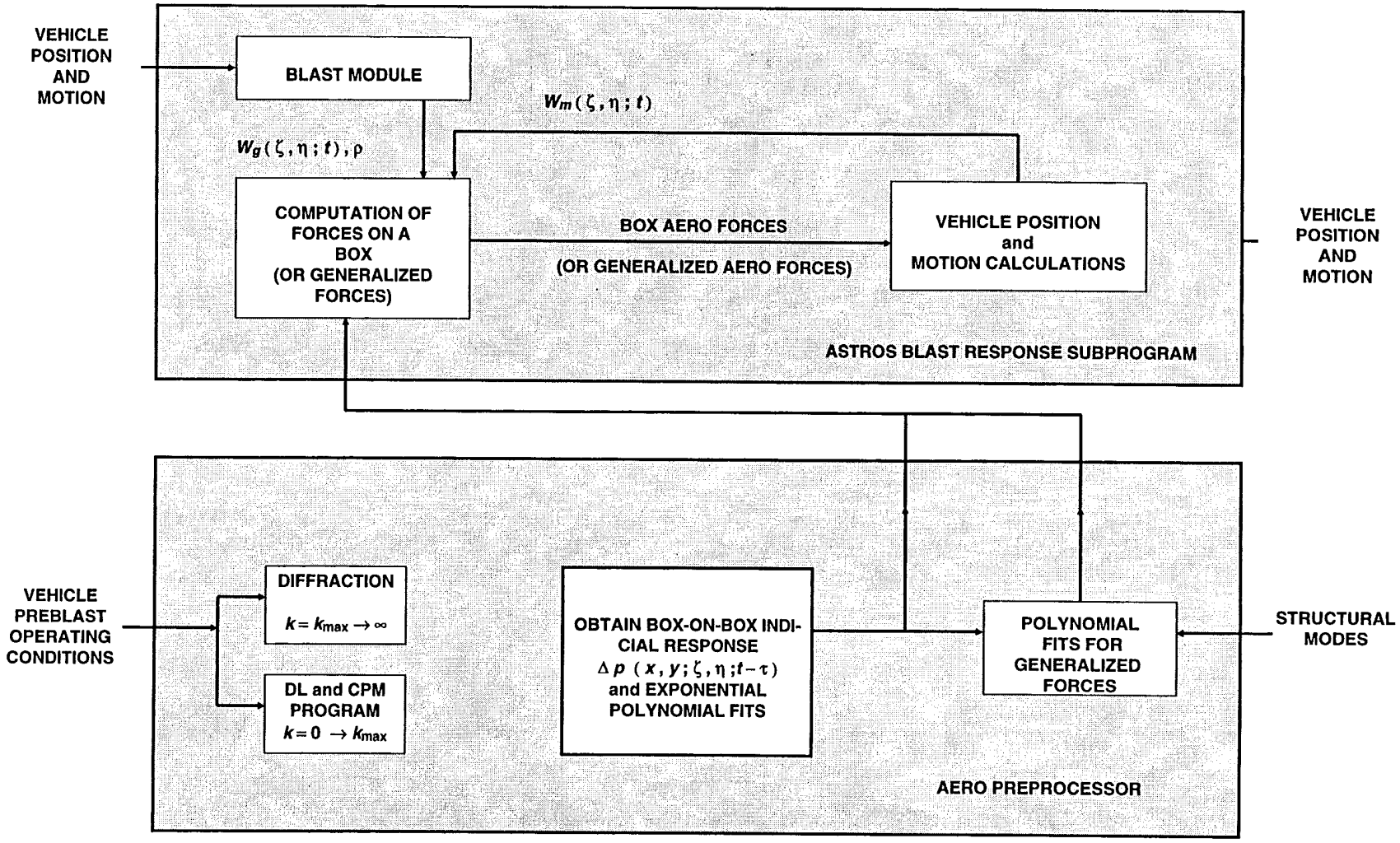


Figure 22. Block Diagram for the Nuclear Blast Calculation

As discussed in Subsection 8.2.2, the unsteady aerodynamic calculations are performed in the frequency domain. This necessitates a transformation to the time domain and this transformation is the subject of Appendix B. If the algorithm of Appendix B is considered a "black box," its inputs are the A matrices of Equation 8-7 at a series of reduced frequencies and its outputs are a new set of matrices that represent the indicial response of a receiving box due to a disturbance at a sending box. Equation B-10 can be written in matrix form as:

$$\boldsymbol{F}(t,t') = \boldsymbol{M}_{ss} + \sum_{n=1}^{N} \boldsymbol{M}_{tt}^{n}\, e^{-\beta_n (t,t')} \tag{12-1}$$

where

$\boldsymbol{F}$	Matrix of forces at receiving points at time t due a unit normal wash at a sending point at time t'.
$\boldsymbol{M}_{ss}$	Matrix of steady-state influence coefficients
$\boldsymbol{M}_{tt}$	Matrices of transient influence coefficients
β_n	Exponential coefficient
N	Number of terms used for transient influence coefficient representation
t	Time
t'	Time of application of the disturbance

To retain physical insight into the calculations, it is necessary to replace the rigid body mode shapes calculated as part of the eigenanalysis with mode shapes that represent the rigid pitch and plunge of the aircraft about the support point. The two sets of modes will be the same if the support point is at the center of gravity of the aircraft, but it must be assumed that the center of gravity location is either not known in a typical application or that there is no grid point at the particular location. The eigenanalysis modes can be represented as

$$\varphi_{EIG} = \left[\varphi_R \mid \varphi_E \right] \tag{12-2}$$

where the R and E subscripts refer to rigid and elastic modes, respectively. The blast analysis replaces the extracted R with new modes:

$$\Phi_{BL} = \left[\frac{\boldsymbol{D}}{\boldsymbol{I}} \mid \Phi_E \right] \tag{12-3}$$

where the $\boldsymbol{D}$ matrix is the rigid body transformation matrix first discussed in Equation 6-17. With these new mode shapes, the generalized mass matrix needs to be recomputed and is likely to have off-diagonal terms for the rigid body modes. The generalized stiffness matrix is unchanged since the terms related to rigid body modes are null.

The computation of the $\boldsymbol{M}_{ss}$ and $\boldsymbol{M}_{tt}$ matrices is the function of the middle box in the preprocessor portion of Figure 22. Each of these matrices is a square matrix with a dimension equal to the number of

boxes in the unsteady aerodynamic model. To make the blast response calculations less demanding of computer storage and CPU usage, a reduction of these matrices to a generalized form is performed:

$$\begin{aligned} \boldsymbol{M}_{ss}^{g} &= \Phi_{BL}^{T}\ \boldsymbol{BG}_{ja}\ \boldsymbol{M}_{ss} \\ \boldsymbol{M}_{tt}^{g} &= \Phi_{BL}^{T}\ \boldsymbol{BG}_{ja}\ \boldsymbol{M}_{tt} \end{aligned} \tag{12-4}$$

where $\boldsymbol{BG}_{ja}$ is the spline matrix. This matrix is derived from the $\boldsymbol{UG}$ matrix of Equation 8-44 by retaining rows that correspond to displacements (i.e., by deleting rows associated with slopes).

12.1. TRIM FOR THE BLAST ANALYSIS

The aircraft starts from a specified maneuver condition which provides initial conditions for a blast response calculation. The trim analysis is much like the one already given for steady aerodynamics analysis in Subsection 6.2.2 with the important distinction that analyses are done in modal coordinates for this case. The governing equation of motion is

$$\boldsymbol{M}_{hh}\,\ddot{\eta} + \left(\boldsymbol{K}_{hh} + \bar{q}\ \boldsymbol{Q}_{hh}\right)\eta = \bar{q}\ \boldsymbol{F}_{R}\ \delta \tag{12-5}$$

The $\boldsymbol{M}$ and $\boldsymbol{K}$ matrices are the generalized mass and stiffness matrices just discussed while Q is the generalized aeroelastic correction matrix, which is calculated using

$$\boldsymbol{Q}_{hh} = \boldsymbol{M}_{ss}^{g}\ \boldsymbol{BGS}_{ja}\ \Phi_{BL} \tag{12-6}$$

where $\boldsymbol{BGS}_{ja}$ is the counterpart of the $\boldsymbol{BG}$ matrix of Equation 12-4 that only contains terms from the $\boldsymbol{UG}$ matrix that provide slopes at the aerodynamic panels.

The $\boldsymbol{F}_{R}$ matrix contains rigid body load vectors for unit values of the angle of attack, pitch rate and trim surface angle, much like the ***AIRFRC*** matrix of Subsection 8.1.2. This matrix is calculated from

$$\boldsymbol{F}_{R} = \boldsymbol{M}_{ss}^{g}\,\boldsymbol{w}_{o} \tag{12-7}$$

where the $\boldsymbol{w}_{o}$ matrix contains the downwash vectors for each of the aerodynamic parameters discussed above. Completing the description of Equation 12-5, η is a vector of generalized coordinates while $\bar{q}$ is the dynamic pressure.

It is necessary to distinguish between rigid body and flexible coordinates in the solution of Equation 12-5:

$$\begin{bmatrix} \boldsymbol{M}_{rr} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_{ee} \end{bmatrix} \begin{Bmatrix} \ddot{\eta}_{r} \\ \ddot{\eta}_{e} \end{Bmatrix} + \begin{bmatrix} \boldsymbol{0} & \bar{q}\,\boldsymbol{Q}_{re} \\ \boldsymbol{0} & \boldsymbol{K}_{ee} + \bar{q}\,\boldsymbol{Q}_{ee} \end{bmatrix} \begin{Bmatrix} \ddot{\eta}_{r} \\ \ddot{\eta}_{e} \end{Bmatrix} = \bar{q} \begin{bmatrix} \boldsymbol{F}_{R}^{r} \\ \boldsymbol{F}_{R}^{e} \end{bmatrix} \delta \tag{12-8}$$

Note that $\boldsymbol{Q}_{rr}$ and $\boldsymbol{Q}_{re}$ are set to zero since the rigid body aerodynamic forces are already contained in the $\boldsymbol{F}_R$ matrix. For the trimmed condition, $\ddot{\eta}_r = 0$ so that the second row of Equation 12-8 can be solved for η_e in terms of δ:

$$\eta_e = \bar{q} \left[\boldsymbol{K}_{ee} + \bar{q}\, \boldsymbol{Q}_{ee} \right]^{-1} \boldsymbol{F}_R^e \, \delta \tag{12-9}$$

This is then substituted into the first row of Equation 12-9 to give

$$\boldsymbol{M}_{rr} \, \ddot{\eta} = \bar{q} \left[\boldsymbol{F}_R^r - \boldsymbol{Q}_{re} \bar{q} \left[\boldsymbol{K}_{ee} + \bar{q}\, \boldsymbol{Q}_{ee} \right]^{-1} \boldsymbol{F}_R^e \right] \delta \tag{12-10}$$

For a trimmed flight condition, the rigid body acceleration vector is known (the pitch acceleration is zero and the plunge acceleration is given by the load factor), as is the pitch rate. The angle of attack and pitch trim surface angle can then be solved for, and Equation 12-9 can be used to calculate the elastic deformations and therefore complete the trim solution. This represents a special case of symmetric trim in which only one pitch control surface can be used and the unknowns are the angle of attack and the deflection of the control surface.

12.2. BLAST RESPONSE

The basic equation used for the blast response can be written as:

$$\boldsymbol{M} \, \ddot{\eta} + \boldsymbol{K} \, \eta = \boldsymbol{F}_{tot} \tag{12-11}$$

This is a gross simplification of the blast response formulation in that the force terms on the right-hand side of this equation are a combination of a number of factors:

$$\boldsymbol{F}_{tot} = \boldsymbol{F}_{grav} + \boldsymbol{F}_{aero} \tag{12-12}$$

where the gravity force can be considered to act on the first one or two modes only

$$\boldsymbol{F}_{grav} = g \begin{Bmatrix} m_{11} \\ m_{21} \\ 0 \\ . \\ . \\ 0 \end{Bmatrix} \tag{12-13}$$

while the aerodynamic force is a combination of blast, aircraft configuration and aircraft motion effects. These transient aerodynamic effects can be computed at time t in terms of the downwash at the current time plus contributions from the discrete changes in the downwash from all previous steps using a Duhamel integral applied to the indicial representation of Equation 12-1:

$$\boldsymbol{F}_{aero}(t) = \boldsymbol{M}^{g}_{ss}\ \boldsymbol{W}(t) + \sum_{t'=0}^{t'=t} \sum_{n=1}^{N} \left[\boldsymbol{M}^{g}_{tt}\right]_n \exp\left[\frac{-\beta_n (t-t')}{\Delta \boldsymbol{W}(t')}\right] \tag{12-14}$$

where $\boldsymbol{W}(t)$ is the downwash at the current time and $\Delta \boldsymbol{W}(t)$ is the change in the downwash at previous time so that

$$\boldsymbol{W}(t) = \sum_{t'=0}^{t'=t} \Delta \boldsymbol{W}(t) \tag{12-15}$$

A recursion relation can be established for the second term of Equation 12-14 by defining

$$\delta \boldsymbol{W}_n(t) = \sum_{t'=0}^{t'=t} e^{-\beta_n (t-t')}\ \Delta \boldsymbol{W}(t) \tag{12-16}$$

Then the aerodynamic force at a later time, $t + \Delta t$, can be expressed as

$$\begin{aligned} \boldsymbol{F}_{aero}(t+\Delta t) &= \boldsymbol{M}^{g}_{ss}\ \boldsymbol{W}(t) + \Delta \boldsymbol{W}(t+\Delta t) \\ &+ \sum_{n=1}^{N} e^{-\beta_n \Delta t} \left[\boldsymbol{M}^{g}_{tt}\right]_n \delta \boldsymbol{W}_n(t) + \left[\boldsymbol{M}^{g}_{tt}\right]_n \Delta \boldsymbol{W}(t+\Delta t) \end{aligned} \tag{12-17}$$

This process is continued using

$$\delta \boldsymbol{W}_n(t+\Delta t) = e^{-\beta_n \Delta t}\ \delta \boldsymbol{W}_n(t)^{-1}\ \Delta \boldsymbol{W}(t+\Delta t) \tag{12-18}$$

The downwash vector $\boldsymbol{W}(t)$ is computed at each time step by forming the dot product, at each aerodynamic box, of the total velocity at the box and the box normal:

$$\boldsymbol{W}_j(t) = \vec{\boldsymbol{V}}_j(t) \cdot \vec{\eta}_j(t) \tag{12-19}$$

where $\vec{\eta}_j$ is a combination of the jig shape of the aircraft plus additional slopes caused by the elastic deformations and control surface deflections. The velocity vector is the sum of four components:

$$\vec{\boldsymbol{V}} = \vec{\boldsymbol{V}}_{BLAST} + \vec{\boldsymbol{V}}_{TRAN} + \vec{\boldsymbol{V}}_{ROT} + \vec{\boldsymbol{V}}_{ELAS} \tag{12-20}$$

where

$\vec{\boldsymbol{V}}_{BLAST}$ velocity caused by the blast
$\vec{\boldsymbol{V}}_{TRAN}$ velocity caused by vehicle translation at a reference point
$\vec{\boldsymbol{V}}_{ROT}$ velocity caused by vehicle rotation at a reference point
$\vec{\boldsymbol{V}}_{ELAS}$ velocity caused by elastic deformations.

The blast velocity is computed by the algorithm given in Reference 28, while the aircraft translation and rotation and the elastic velocities at the box locations are all computed from initial conditions and the solution of Equation 12-11.

The Newmark-Beta procedure described in Subsection 11.3.2 is used to perform the numerical integration of Equation 12-11 as well. There is no provision for a modal damping matrix in this case so that the definitions of the $\boldsymbol{A}$ and $\boldsymbol{D}$ matrices in Equation 11-35 are somewhat simplified. A final comment is that the loading for the transient response calculation is a function of the displacement in this case, while the formulation of Subsection 11.3.2 has assumed that the loading is not affected by the displacements. Extended testing of the algorithm is required to determine whether this approximation is adequate for this response calculation.

THIS PAGE INTENTIONALLY LEFT BLANK

13. AUTOMATED DESIGN

The role of automated design in ASTROS is to apply resizing algorithms to drive the design toward one that satisfies user-specified criteria in an optimal manner. Section 4 has discussed the multidisciplinary optimization task of ASTROS in terms of the problem. From Section 4, the overall design task is specified as:

Find the set of design variables, v, which will minimize

$$F(v) \tag{4-1}$$

Subject to:

$$g_j(v) \leq 0.0 \qquad j = 1, \ldots ncon \tag{4-2}$$

$$v_i^{lower} \leq v_i \leq v_i^{upper} \qquad i = 1, \ldots ndv \tag{4-4}$$

This section discusses two alternative methods for solving this task: ***mathematical programming*** and ***fully stressed design***. These techniques are complementary in the sense that mathematical programming techniques are quite general in the problems they can solve, but are computationally intensive. Fully stressed design provides an efficient means to solve large design tasks, but this technique is limited to problems subject only to stress constraints. Although only these two methods are present in ASTROS at this time, it is recognized that there are other algorithms that could be used to perform the automated design task. Notably, there are a number of algorithms which can be thought of as representing a synthesis between mathematical programming methods and physical optimality criteria, such as fully stressed design. These further methods could be classified as mathematical optimality criteria methods in that they base their redesign on mathematical criteria that are known to hold true at the optimum. References 29, 30 and 31 contain algorithms that fit in this category although they are quite distinct from one another. These alternatives are not discussed here, but their potential for performing automated design is recognized with further research required to specify exactly how they fit into the ASTROS environment.

13.1. MATHEMATICAL PROGRAMMING

Mathematical programming techniques can be characterized as ***search techniques*** which progress toward an optimum based on information available for the current design. A variety of algorithms are available to perform this search. The μ-DOT algorithm of References 32 and 33 is used in ASTROS. This algorithm combines features from ***feasible directions*** (Reference 34) and ***generalized reduced gradient*** (Reference 35) algorithms to provide an efficient and powerful overall procedure. The μ-DOT algorithm can be characterized as a ***direct method*** in that constraint information is used directly in the

optimization process. Indirect methods, such as the interior penalty function method, the constraints are adjoined to the objective and then an unconstrained optimization procedure is applied.

Optimization algorithms can also be partially characterized by the method they employ in the one-dimensional search that is required to determine the distance to be traveled along a direction that has been determined to give an improved design. The μ-DOT algorithm employs a technique wherein bounds on the move direction are first determined and a polynomial interpolation technique is used to find the minimum within these bounds.

As mentioned, the generality of mathematical programming algorithms is offset by the amount of computer resources required in their application. The remainder of this subsection discusses techniques that are employed in ASTROS to minimize the size of the optimization task, to wrest the maximum amount of usefulness out of each analysis of a particular design and to find the balance between performing too many structural analyses and too few. Reference 6 provided the basis for many of the concepts discussed here.

13.1.1. Reduction of the Number of Design Variables

As discussed in Subsection 4.2.1, design variable linking is used to permit the application of mathematical programming algorithms to practical structural design problems. There is no fixed limit on the number of design variables a mathematical programming algorithm can handle in general, nor does ASTROS, in particular, impose any limits. The limits are indirect in that computer resource requirements are a nonlinear function of the number of design variables and constraints. Experience has indicated that problems with two to three hundred design variables approach the practical limit of problem size that can be attempted. This limit is both subjective, in the sense that different investigators have different tolerances for what they will endure, and machine dependent, with a supercomputer just becoming effective at about the same point that a microcomputer or a workstation is becoming untenable.

13.1.2. Reduction of the Number of Constraints

The number of constraints given by Equation 4-2 that are generated in ASTROS for even a moderate problem can number in the thousands. Each finite element can generate one or more constraints for each load case, while the remaining constraints of Subsection 4.2.2 also contribute to the total number. Typically, only a small percentage of these constraints will affect the final design and it is necessary to exploit this fact, both to reduce the size of the mathematical programming task and to limit the effort required to compute the constraint sensitivities. The basic concept is to retain only those constraints for the design task that could play an active role in the design process. The selection of these critical constraints requires making a judgment, but one with minimal risk if the retention criteria are sufficiently broad. Two retention criteria are applied in ASTROS:

- All constraints with a value greater than a specified value, ε, are retained.
- The most critical **NRFAC** x *ndv* constraints are always retained, where **NRFAC** is a user specified parameter and *ndv* is the number of global design variables.

Default values of **NRFAC** = 3.0 and $\varepsilon = -0.10$ are specified in the standard MAPOL sequence. These values can be tailored to a specific application by editing this standard sequence.

When shape functions are used, certain user-specified thickness constraints data entries (See Subsection 4.2.2.3) are retained in addition to those previously selected. This is because the retention criteria are not adequate to predict whether these crucial constraints will drive the design. If they are not retained and become violated during the redesign process, the design can be driven to physically unrealistic values that would make further analyses incorrect.

Following the determination of the active constraints, a screening operation takes place in each boundary condition that can provide significant efficiencies if some of the operations performed during the analysis phase do not require sensitivity calculations. For example, for a run with multiple boundary conditions, one or more of the boundary conditions may not contain any active constraints. In this case, there would be no need to process the inactive boundary condition(s). Further, within a boundary condition, certain disciplines may contain active constraints while others do not. Again, the inactive disciplines do not require further processing. Finally, within a discipline, some subcases may not contain active constraints and therefore are not included in the sensitivity evaluation.

13.1.3. The Approximate Design Problem

Once information on the current design is obtained, it is passed to the μ-DOT procedure for processing seven basic pieces of information:

F_o the current value of the objective

v_o vector of current values of the design variables

g_o vector of current values of the active constraints

r_o vector of current values of the active user functions responses

$\frac{\partial F_o}{\partial v_i}$ vector of gradients of the objective with respect to the design variables

A Matrix of the gradients of the active constraints with respect to the design variables $\frac{\partial g_o}{\partial v_i}$

A_R^u Matrix of gradients of active responses associated with user functions $\frac{\partial r_o}{\partial v_i}$

where the o subscript indicates that quantities have been calculated for the current value.

Since the analysis phase of ASTROS is the most costly, it is important to minimize the number of complete analyses that are performed. This is done by performing the redesign under the assumption that the gradients are invariant with respect to changes in the design variable. This is equivalent to performing a first order Taylor series expansion about the original design and using this information in the redesign. It is, therefore, very important that the gradient information be of high quality. As discussed in Reference 6, one way of ensuring this quality is to consider the physical nature of the constraint and, in particular, to recognize that stress and strain are nearly linearly proportional to the inverse of the

physical design variables (for determinate structures, the linear relation is exact). It is for this reason that ASTROS, in the case of unique linking and physical linking (see Subsection 2.2.1) defines a new variable that is the inverse of the global variable:

$$x_i = \frac{1}{v^i} \tag{13-1}$$

In terms of the direct gradient information that is computed by ASTROS, a direct or inverse Taylor Series expansion of the desired function is formed:

$$\bar{f} = f_o - \sum_{i=1}^{ndv} \frac{\frac{\partial f}{\partial v_i}\left(x_i - x_{oi}\right)}{x_{oi}^2} \quad \text{(inverse approximation)} \tag{13-2a}$$

$$\bar{f} = f_o + \sum_{i=1}^{ndv} \frac{\partial f}{\partial v_i}\left(v_i - v_{oi}\right) \quad \text{(direct approximation)} \tag{13-2b}$$

The μ-DOT procedure requires the evaluation of the objective and constraint functions and their gradients. The gradients are not invarient in the case of inverse variables and for user function constraints, as will be presented. To re-evaluate the function gradients, the following expressions are used:

$$\frac{\overline{\partial f}}{\partial x_i} = -\frac{1}{x_{o_i}^2}\frac{\partial f}{\partial v_i} \quad \text{(inverse approximation)} \tag{13-3a}$$

$$\frac{\overline{\partial f}}{\partial v_i} = \frac{\partial f}{\partial v_i} \quad \text{(direct approximation)} \tag{13-3b}$$

For the objective function and the set of canonical ASTROS constraints, the constraint or objective function is directly approximated using Equation 13-2 or 13-3 as appropriate — the choice is made by ASTROS to exploit the known characteristics of the design variables and constraints.

For the constraint gradient sensitivity calculation of Equation 13-3, μ-DOT makes its own determination as to which constraints are expected to be active during the design and requests gradient information only for this reduced set. This results in a slight efficiency in terms of the calculations required by Equation 13-3; more importantly, the efficiency of the μ-DOT procedure is strongly affected by the number of constraints it retains.

The relations given by Equations 13-2 and 13-3 are high quality approximations. The fact that this does entail approximations is recognized by imposing constraints on the movement of the inverse or direct design variables:

$$\frac{x_{o_i}}{\mathbf{MOVLIM}} \le x_i \le \mathbf{MOVLIM} \cdot x_{o_i} \quad \text{(inverse approximation)} \tag{13-4a}$$

$$\frac{v_{o_i}}{\texttt{MOVLIM}} \le v_i \le \texttt{MOVLIM} \cdot v_{o_i} \quad \text{(direct approximation)} \tag{13-4a}$$

where a default value of **MOVLIM** = 2.0 is specified in the standard MAPOL sequence and can be changed by the user. As used by ASTROS, **MOVLIM** must always be greater than 1.0. As a final comment on move limits, if the upper and lower bounds specified by Equation 13-4 exceed user specified values on the variable

$$\frac{1}{v_i^{max}} \le x_i \le \frac{1}{v_i^{min}} \tag{13-5a}$$

$$v_i^{min} \le v_i \le v_i^{max} \tag{13-5b}$$

the user specified values are used as the side constraints in μ-DOT. The $v_i^{\min}$ and $v_i^{\max}$ are user input values for the maximum and minimum allowed values for the i^{th} direct design variable.

If shape function linking is used, the inverse design variable concept cannot be used. The physical significance of using the inverse variable is not clear in this case, but more importantly, the design variable values can pass through zero so that the inverse variable would be infinite. The function and gradient evaluations for this case are then formed using direct approximation.

For user function constraints or objective function, a slightly different approach to building the approximation is taken. This is best illustrated by reviewing the function form:

$$a_s = h(r_1, r_2, \dots, r_n) \tag{4-59}$$

where:

a_s	is the synthetic function value
h	is some set of algebraic operators
r_i	are responses of the finite element analysis and/or model characteristics

In general, the derivative of the user function is:

$$\frac{\partial a_s}{\partial v_j} = \bar{h}\left(r_1, \frac{\partial r_1}{\partial v_j}, r_2, \frac{\partial r_2}{\partial v_j}, \dots, r_n, \frac{\partial r_n}{\partial v_j}\right) \tag{4-60}$$

where $\bar{h}$ is determined automatically by ASTROS from the symbolic differentiation of Equation (4-59). For purposes of the approximate problem, a_s and $\frac{\partial a_s}{\partial v}$ must be re-evaluated when directed by μ-DOT. To perform this re-evaluation, ASTROS approximates a_s as:

$$\overline{a_s} = h(\overline{r_1}, \overline{r_2}, \dots, \overline{r_n}) \tag{13-6}$$

where $\overline{r_i}$ is obtained using r_o and A_k^u in Equation (13-2). Direct or inverse approximation is used depending on the nature of r_i.

Similarly, $\frac{\partial a_s}{\partial v}$ is approximated from:

$$\overline{\frac{\partial a_s}{\partial v_j}} = \overline{h}\left(\overline{r_1}, \frac{\partial \overline{r_1}}{\partial v_j}, \overline{r_2}, \frac{\partial \overline{r_2}}{\partial v_j}, \ldots, \overline{r_n}, \frac{\partial \overline{r_n}}{\partial v_j}\right) \tag{4-60}$$

where $\overline{r_i}$ and $\frac{\partial \overline{r_i}}{\partial v}$ are obtained using r_o and A_k^u in Equation (13-2) and (13-3). This approach preserves the nonlinearities that are intrinsic to the user's function, h, while allowing an efficient re-evaluation of an approximate a_s during the actual resizing step. Again, the move limit controls are used to prevent the design process from straying due to errors in the approximations.

13.1.4. Termination Criteria

The decision as to when to terminate an automated design procedure is a subjective one. The goal is to find a balance between premature termination before the design has converged on the one hand and performing wasteful iterations after the design has, for all practical purposes, reached an optimum on the other hand. Termination criteria are imposed at two levels within the ASTROS procedure. The first is within the redesign phase of Figure 1 and uses μ-DOT criteria to terminate this phase. Since μ-DOT has solved an approximate problem, a number of iterations are required to find a converged optimum. For the second level, it is therefore necessary, following each redesign, to check whether the design can be considered converged.

The termination criteria within μ-DOT are based on changes in the objective function. If the absolute value of the change in the objective function is less than μ-DOT parameter DABOBJ or the relative change in the objective function is less than μ-DOT parameter DELOBJ for ITRMOP iterations, the μ-DOT procedure is terminated. Default parameters, which may be overridden by bulk data input, for the three parameters are DABOBJ = 0.001 F_o, DELOBJ = 0.001 and ITRMOP = 2. F_o is the initial value of the objective when μ-DOT is invoked.

Following the μ-DOT redesign, an initial determination is made as to whether the design has converged. The criteria used here is similar to the μ-DOT criteria in that the design is tentatively judged to be converged if:

$$| \Delta F | \leq 0.005 \tag{13-13}$$

or if:

$$\left| \frac{\Delta F}{F_o} \right| \leq 0.01 \ \texttt{CNVLIM} \tag{13-14}$$

where ΔF is the change in the objective for the current redesign and **CNVLIM** is defined in the standard MAPOL sequence to be 1.0. Equation 13-14, therefore specifies convergence when less than a 1.0 percent change is made in the weight of the structure.

So far, the discussion has been in terms of changes in the objective function, but clearly the values of the constraints have to be considered before a final convergence determination can be made. If an initial test of Equation 13-13 or 13-14 is satisfied, it is necessary to make a further analysis of the redesigned structure to see if all the constraint conditions are satisfied. These constraints could be violated because μ-DOT was not able to achieve a feasible design based on the information given to it. Alternatively, all the constraints of the approximate problem given by Equation 13-5 or 13-9 may be satisfied, but a reanalysis may find them violated. Final convergence is determined to have occurred when one of the conditions of Equation 13-13 or 13-14 is satisfied and the largest constraint value, following reanalysis, satisfies

$$2.0 \cdot \mathbf{CTL} < g_{\max} \leq 3.0 \cdot \mathbf{CTLMIN} \tag{13-15}$$

where **CTL** is a μ-DOT parameter used to designate whether a constraint is active and **CTLMIN** is a μ-DOT parameter that is used to designate whether a constraint is violated. These parameters are initially set by the user or to default value of **CTL** = - 0.003 and **CTLMIN** = 0.0005. μ-DOT can reduce these numbers further as part of the optimization process so that the criteria of Equation 13-15 are quite stringent. The factors of two and three given in Equation 13-15 allow for some leeway in differences caused by the approximation to the constraints.

Note that a lower bound limit is applied in Equation 13-15 to avoid the case of the procedure termination when there are no active constraints. If none of the constraints are active, the current design is not optimal. The final ASTROS design termination criteria is based on the number of analysis cycles that have been made. This criteria is imposed to safeguard against the case where the redesign process is unable to converge. It can also be used to limit the number of iterations that are made when there is uncertainty as to whether the design problem has been properly posed. The default value for the maximum iterations is **MAXITER** = 15, with experience indicating that termination rarely occurs because this number is exceeded.

13.2. FULLY STRESSED DESIGN

A Fully Stressed Design (FSD) resizing option has been provided in ASTROS to complement the standard mathematical programming optimization methods. While ASTROS is primarily a multidisciplinary optimization tool and FSD methods are, by definition, severely limited in scope, this method was included because of its rapidity in achieving a feasible strength design and because it represents a relatively well known optimization method. The implementation of FSD in ASTROS recognizes the inherent limitations of this method, however, and no attempt was made to make this option handle the full range of optimization problems that ASTROS supports. Instead, the FSD option is intended to be used as a preliminary step to achieve a feasible or near optimal strength design from which to continue the optimization using the more general methods. It is, of course, useful in its own right for problems in which only stress constraints for static disciplines are applied.

The utility of the FSD resizing option in ASTROS is that, for problems where static strength constraints play an important role in determining the structural sizes, FSD can find a reasonable initial design very quickly. Therefore, while the FSD method itself can treat only the static stress constraints, the FSD option may be used in almost any optimization problem in ASTROS where stress constraints are applied. There is only one restriction to the use of FSD: it cannot be used in combination with shape function design variable linking. This restriction is discussed further in Subsection 13.2.2. Since the determination of an initial design is the typical purpose for FSD in ASTROS, the algorithm has been implemented in such a way that the user selects some number of initial design cycles to be performed using FSD. After these cycles have been completed, ASTROS automatically reverts to mathematical programming methods until convergence or the maximum number of iterations is reached. The user who wishes to use only FSD methods can easily direct that all iterations use the FSD option.

13.2.1. The FSD Algorithm for Local Design Variables

In the ASTROS implementation of the FSD resizing concept, the new local design variable (which represents the physical property of one finite element; e.g., the thickness of a shear panel) is found based on the ratio of stress to the allowable stress:

$$t_{i_{new}} = \max\left[\left(\frac{\sigma}{\sigma_{all}}\right)_i^{\alpha} \quad t_{i_{old}}, t_{i_{min}}\right] \tag{13-16}$$

The stress ratio $\frac{\sigma}{\sigma_{all}}$ is determined in ASTROS from the applied von Mises and/or Tsai-Wu stress constraints. These constraints have been formulated such that:

$$\left(\frac{\sigma}{\sigma_{all}}\right)_i = g_i + 1.0 \tag{13-17}$$

where g_i represents the current stress constraint value. By substituting Equation 13-17 into 13-16, it is possible to very quickly determine a new set of local design variables. The only difficulty in performing this operation is in the bookkeeping to determine which stress constraint corresponds to a particular local design variable. More important is the treatment of stress constraints applied to undesigned elements for which there is no corresponding local variable. Since the ASTROS implementation of this method is intended to be approximate, we decided to ignore these stress constraints in the computation of the new local design variable vector. This is consistent with the fact that all the other constraint types are also ignored.

To offer an improved convergence behavior for this FSD algorithm, the exponential factor, α, has been provided in Equation 13-16. Small values of α result in better convergence at the expense of additional iterations. The value of this parameter is user selectable in ASTROS, but defaults to 0.90. This value was chosen for its rapid movement toward a fully stressed design in the initial iterations. If FSD is intended to be used to achieve a final converged solution, a value of 0.50 or less is preferred.

13.2.2. Global Design Variable Determination

The local design variables, $\boldsymbol{t}$, may be linked in ASTROS to the global design variables, $\boldsymbol{v}$, through a number of options described in Subsection 4.3. After the new set of local variables have been determined using the algorithm described in the preceding subsection, an additional step is required to determine the new set of global design variables. The method of determining the new global variables is based on the linear linking relationship:

$$\boldsymbol{t} = \boldsymbol{P}\boldsymbol{v} \tag{2-6}$$

In the unique linking and physical linking options in ASTROS, each local variable is uniquely associated with one global variable, although a global variable may control many local variables. In these cases, a set of global variables is found from the new local variables by using the following:

$$v_{j_{new}} = \left[\max \frac{t_{i_{new}}}{P_{ij}} \right] \qquad \text{over all nonzero terms in the } j^{th} \text{ column of } \boldsymbol{P} \tag{13-18}$$

This determines a conservative set of global variables which satisfy the resizing as defined in Equation 13-16.

In the third, shape function, linking option in ASTROS, a single local design variable may be controlled by many global design variables. Therefore, there is no straightforward method to determine the optimal set of global design variables to satisfy the linking relationship of Equation 2-6. While such a determination could be made, the current implementation of FSD in ASTROS does not support this linking option. In such cases, the ASTROS procedure will automatically revert to mathematical programming methods.

THIS PAGE INTENTIONALLY LEFT BLANK

14. REFERENCES

1. MacNeal, R.H., *The NASTRAN Theoretical Manual*, NASA SP-221(01), April 1971.
2. _____, *The NASTRAN Programmer's Manual*, NASA SP-223(04), December 1977.
3. _____, *The NASTRAN User's Manual*, NASA SP-222(06), December 1983.
4. Lynch, R.W., Rogers, W.A., Brayman, G.W., and Hertz, T.J., "Aero-elastic Tailoring of Advanced Composite Structures for Military Aircraft, Volume III - Modifications and User's Guide for Procedure TSO," AFFDL-TR-76-100, Volume III, February 1978.
5. Markowitz, J. and Isakson, G., "FASTOP-3: A Strength, Deflection and Flutter Optimization Program for Metallic and Composite Structures," AFFDL-TR-78-50, Volumes I and II, May 1978.
6. Schmit, L.A., Jr. and Miura, H., "Approximation Concepts for Efficient Structural Synthesis," NASA CR-2552, March 1976.
7. Sobieszczanski - Sobieski, J., James, B.B., and Dovi, A.R., "Structural Optimization by Multilevel Decomposition," AIAA Journal, Volume 23, No. 11, November 1985, pp 1775-1782.
8. Tsai, S.W. and Hahn, H.T., *Introduction to Composite Materials*, TECHNOMIC Publishing Co., Inc., Westport, CT, 1980.
9. Bisplinghoff, R.L., Ashley, H., and Halfman, R.L., *Aeroelasticity*, Addison-Wesley Publishing Co., Inc., Reading, Massachusetts, 1955, pp 454-469.
10. Herendeen, D.L., Hoesly, R.L., and Johnson, E.H., "Automated Strength-Aeroelastic Design of Aerospace Structures," AFWAL-TR-85-3025, September 1985.
11. Herendeen, D.L., Hoesly, R.L., Johnson, E.H., and Venkayya, V.B., "ASTROS - An Advanced Software Environment for Automated Design," presented at the AIAA/ASME/ASCE/AHS 27th Structures, Structural Dynamics and materials Conference, AIAA-86-0856, San Antonio, Texas, May 1986.
12. Hoesly, R.L., "ASTROS Data Base Specifications," Universal Analytics, Inc., Internal Report, 1984.
13. Date, C.J., *An Introduction to Data Base Systems*, Addison-Wesley Publishing Company, 1975.
14. Schaeffer, H.G., *MSC/NASTRAN Primer, Statics and Normal Modes Analysis*, Schaeffer Analysis, Inc., Mont Vernon, New Hampshire, 1982.
15. Gockel, M.A., Editor, *MSC/NASTRAN Handbook for Dynamic Analysis*, The MacNeal-Schwendler Corporation, Pasadena, California, 1987.
16. Rodden, W.P., Editor, *MSC/NASTRAN Handbook for Aeroelastic Analysis*, The MacNeal-Schwendler Corporation, Pasadena, California, 1987.
17. Arora, J.S. and Haug, E.J., "Methods of Design Sensitivity Analysis in Structural Optimization," AIAA Journal, Volume 17, No. 9, September 1979, pp 970-974.
18. Vanderplaats, G.N., "Comment on Methods of Design Sensitivity Analysis in Structural Optimization," AIAA Journal, Volume 18, No. 11, November 1980, pp 1406-1407.
19. Joseph, Jerrard, A., Ed., *MSC/NASTRAN Application Manual*, the MacNeal-Schwendler Corporation, Pasadena, California, 1986.

20. Wilkinson, J.H., *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, England, 1965.

21. Fox, R.J. and Kapoor, M.P., "Rates of Change of Eigenvalues and Eigenvectors," AIAA Journal, Volume 6, No. 12, December 1968, pp 2426-2429.

22. Woodward, F.A., "An Improved Method for the Aerodynamic Analysis of Wing-Body-Tail Configurations in Subsonic and Supersonic Flow, Part I - Theory and Applications," NASA CR-2228, May 1973.

23. Appa, K. and Yamane, J.R., "Update Structural Design Criteria, Design Procedures and Requirements for Fighter Type Airplane Wing and Tails - Volume I. Nonlinear Maneuver Loads Analysis of Flexible Aircraft: MLOADS Theoretical Development," AFWAL TR-82-3113, Volume I, May 1983.

24. Albano, E. and Rodden, W.P., "A Doublet-Lattice Method for Calculating Lift Distributions on Oscillating Surfaces in Subsonic Flows," AIAA Journal, Volume 7, February 1969, pp 279-285, and Volume 7, November 1969, p 2192.

25. Appa, K., "Constant Pressure Panel Method for Supersonic Unsteady Air-loads Analysis," Journal of Aircraft, Volume 24, October 1987, pp 696-702.

26. Appa, K. and Smith, M.J.C., "Evaluation of the Constant Pressure Panel Method (CPM) for Unsteady Air Loads Prediction," presented as paper AIAA-88-2282 at the AIAA/ASME/ASCE/AHS 29th Structures, Structural Dynamics and Materials Conference, Williamsburg, Virginia, April 1988.

27. Harder, R.L. and Desmarais, R.N., "Interpolation Using Surface Splines," Journal of Aircraft, Volume 9, No. 2, February 1972.

28. Lee, W., "MORE - Moving Receiver Environment Program," Kaman AviDyne Report KA-TR-238, June 1987.

29. Venkayya, V.B., "A Further Generalization of Optimality Criteria Approaches in Structural Optimization," presented at the International Conference on Computational Engineering Science, Atlanta, Georgia, April 1988.

30. Rizzi, P., "Optimization of Multi-Constrained Structures Based on Optimality Criteria," contained in the Proceedings of the AIAA/ASME/ SAE 17th Structures, Structural Dynamics and materials Conference, King of Prussia, Pennsylvania, May 1976, pp 448-462.

31. Fleury, C. and Schmit, L.A., Jr., "Dual Methods and Approximation Concepts in Structural Analysis," NASA CR 3226, 1980.

32. Vanderplaats, G.N., "An Efficient Feasible Directions Algorithm for Design Synthesis," AIAA Journal, Volume 22, No. 11, November 1984, pp 1633-1640.

33. "MICRO-DOT User's Manual, Version 1.0," Engineering Design Optimization, Inc., Santa Barbara, California, 1985.

34. Vanderplaats, G.N. and Moses, F., "Structural Optimization by Methods of Feasible Directions," Journal of Computers and Structures, Volume 3, July 1973, pp 739-755.

35. Gabriele, G.A. and Ragsdell, K.M., "Large Scale Nonlinear Programming Using the Generalized Reduced Gradient Method," Journal of Mechanical Design, Volume 102, No. 3, July 1980, pp 566-573.

36. Press, William H., et al, *Numerical Recipes: The Art of Scientific Computing* Section 3.5 Coefficients of the Interpolating Polynomial, Cambridge University Press, New York, 1986.

APPENDIX A. THE QUAD4 ELEMENT

This appendix provides the theoretical development for the QUAD4 element that has been installed into ASTROS. An overview of this element is given in Subsection 5.3.3, while this appendix provides detailed information on the element. This detail is necessary because, unlike the other elements, the ASTROS QUAD4 element has not been documented elsewhere.

A.1. DISPLACEMENT FUNCTIONS

The QUAD4 element has two distinct element coordinate systems. These are the "user defined" element coordinate system as defined by the element connectivity data and the "internal element" coordinate system, which is defined as having its origin at $G_o \equiv (X_E^o, Y_E^o, Z_E^o)$. This origin is computed by taking the average of the grid point coordinates. The positive X- and Y-axes of the internal element coordinate system are defined with the aid of two points, G_{x_E} and G_{y_E} described below.

V_{13} and V_{24} are defined as the unit diagonal vectors as illustrated in Figure A-1. Thus, the coordinates of points G_{x_E} and G_{y_E} are given by the following:

$$G_{X_E} = \left\{ (X_E^o + X'_E),\ (Y_E^o + Y'_E),\ (Z_E^o) \right\}$$
$$G_{Y_E} = \left\{ (X_E^o - Y'_E),\ (Y_E^o + X'_E),\ (Z_E^o) \right\} \qquad \text{(A-1)}$$

where, X_E^o, Y_E^o and Z_E^o are the coordinates of the origins of the internal coordinate system and X'_E and Y'_E are the components of the bisector vector of the unit diagonals V_{13} and V_{24}.

The coordinates of points G_o, G_{x_E} and G_{y_E}, are used to define the transformation from the internal element coordinate system to the coordinate system in which the grid points are defined. The internal element coordinate system is necessary to correctly handle irregular-shaped and non-planar elements and is henceforth referred to as the *element* (*E*) coordinate system.

Using 2-D interpolation functions, the geometry field at any point (ξ, η) in the element cross-section (see Figure A-2) is defined, where the nodal curvilinear coordinates are related to the nodal cartesian coordinate system in the element coordinate system by the following relationship:

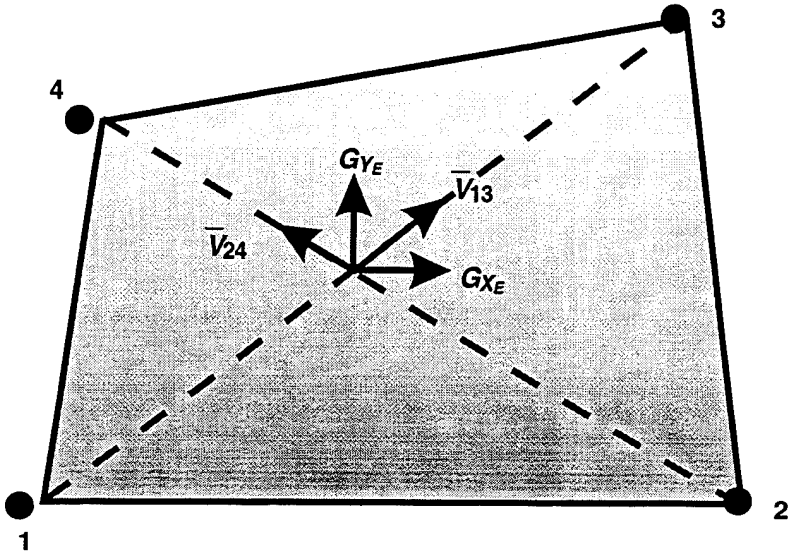


Figure A-1. Internal Element Coordinate System

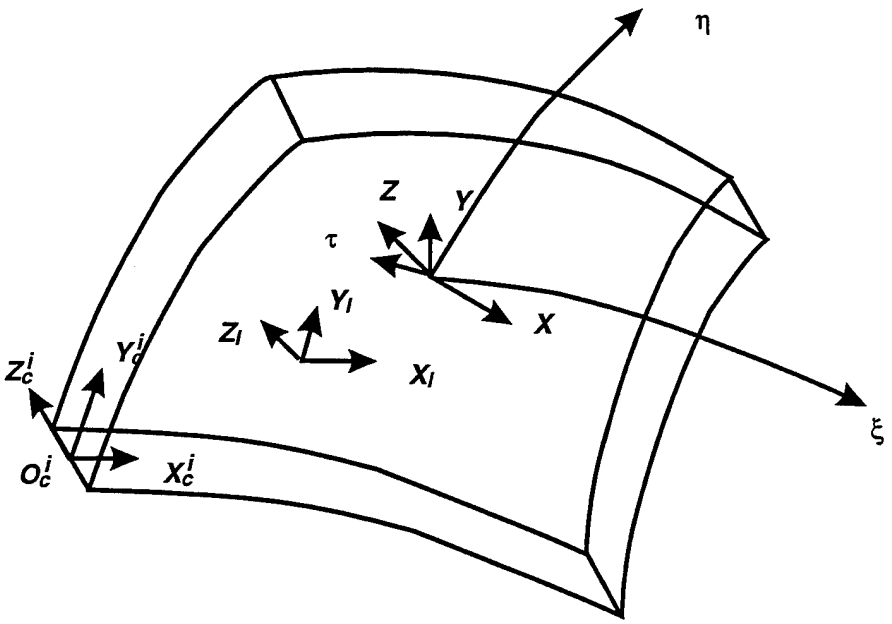


Figure A-2 Isoparametric Quadrilateral 4-Node

$$\boldsymbol{X}_E\ (\xi,\ \eta) = \sum_{i=1}^{4} N_i\ (\xi,\ \eta)\ \boldsymbol{X}_E^i$$

where i refers to grid point i, and

$$\boldsymbol{X}_E^i = \left\{ \boldsymbol{X}_E\ ,\ \boldsymbol{Y}_E\ ,\ \boldsymbol{Z}_E \right\} \text{ at node } i,$$

$N\ (\xi,\ \eta)$ are the interpolation (shape) functions which define the contribution of each node at a given point with the element. These functions and their derivatives are:

$$N_i = \frac{1}{4}(1+\xi\xi_i)\ (1+\eta\eta_i)$$

$$\frac{\partial N_i}{\partial \xi} = \frac{1}{4}\eta_i\,(1+\xi\xi_i) \qquad \text{(A-2)}$$

$$\frac{\partial N_i}{\partial \eta} = \frac{1}{4}\xi_i\ (1+\eta\eta_i)$$

The deformations of the element are also represented with the identical interpolation functions:

$$\boldsymbol{U}_E\ (\xi,\ \eta) = \sum_{i=1}^{4} N_i\ (\xi,\ \eta)\ \boldsymbol{U}_E^i \qquad \text{(A-3)}$$

where $\boldsymbol{U}_E^i = \left[U_E,\ V_E,\ W_E,\ \theta_{x_E},\ \theta_{y_E},\ \theta_{z_E} \right]^T$ represents the vector of displacements at grid point i in the element coordinate system.

A.2. STRAIN-DISPLACEMENT RELATIONSHIP

The QUAD4 element incorporates a reduced solid theory for thick shells. According to this theory, the element has five dof at each grid, defined in a coordinate system whose X-Y plane is tangent to the mid-surface of the shell at the given grid point. The z-axis, therefore, is the normal to mid-surface at that point. In our nomenclature, this is called the C system (Figure A-2 and A-3).

A generalization of the C system, called I system, incorporates the characteristics of the C system at a general point on the mid-surface of the shell element, normally the integration point (Figure A-2).

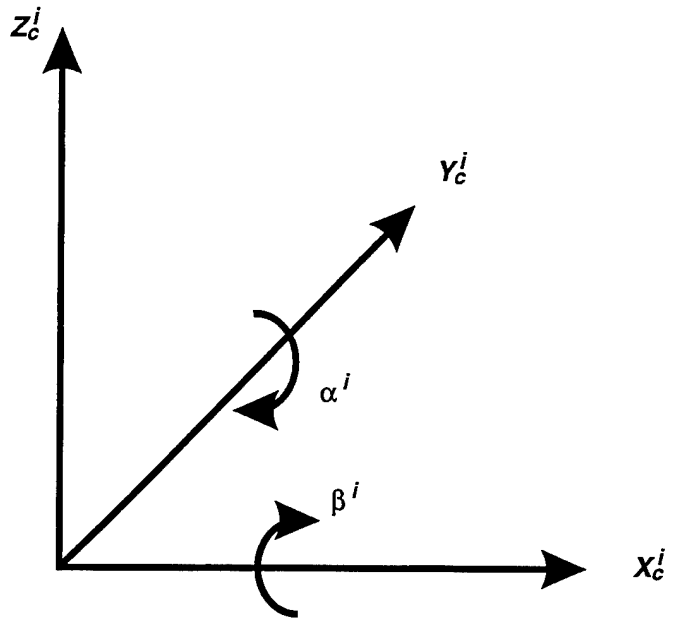


Figure A-3. Deformations at Grid Point i

In order to establish a common definition for I and C systems, consider the following steps:

A. The tangents to mid-surface at a given point (ξ, η) are:

$$V_{t_1} = \frac{\partial \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_E}{\partial \xi} = \sum_{i=1}^{4} \frac{\partial N_i}{\partial \xi} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_E^i \tag{A-4}$$

$$V_{t_2} = \frac{\partial \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_E}{\partial \eta} = \sum_{i=1}^{4} \frac{\partial N_i}{\partial \eta} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_E^i \tag{A-5}$$

where $\begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_E^i$ are the coordinates of grid points in E system.

B. The axes of the new system then follow:

$$\boldsymbol{Z}_{\frac{I}{C}} = \boldsymbol{V}_n = \frac{\boldsymbol{V}_{t_1} \times \boldsymbol{V}_{t_2}}{\left|\boldsymbol{V}_{t_1} \times \boldsymbol{V}_{t_2}\right|}$$

$$\boldsymbol{X}_{\frac{I}{C}} = \frac{\boldsymbol{Y}_E \times \boldsymbol{Z}_{\frac{I}{C}}}{\left|\boldsymbol{Y}_E \times \boldsymbol{Z}_{\frac{I}{C}}\right|} \tag{A-6}$$

$$\boldsymbol{Y}_{\frac{I}{C}} = \boldsymbol{Z}_{\frac{I}{C}} \times \boldsymbol{X}_{\frac{I}{C}}$$

C. Finally:

$$\boldsymbol{T}_{IE} = \left[\boldsymbol{X}_I \; \boldsymbol{Y}_I \; \boldsymbol{Z}_I\right]^T \tag{A-7}$$

$$\boldsymbol{T}_{CE}^i = \left[\boldsymbol{X}_c^i \; \boldsymbol{Y}_c^i \; \boldsymbol{Z}_c^i\right]^T \tag{A-8}$$

Note that the C system is not necessarily invariant when we go from one grid to the next. This is due to the possible warping of the element.

Since the ultimate goal of this discussion is to establish a relationship between the element strains (which are defined in the I system), and the nodal displacements (defined in the E system), it is necessary to develop a series of transformations along with the strain-displacement relationships.

Consider the five degrees of freedom in the C system at each grid point i to be arranged in the following manner (Figure A-3):

$$\boldsymbol{U}_c^i = \begin{Bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \\ \boldsymbol{w} \end{Bmatrix}_c^i \quad ; \quad \theta_c^i = \begin{Bmatrix} \alpha \\ \beta \end{Bmatrix}_c^i \tag{A-9}$$

In order to be compatible with the other DOF in the model, these are related to the six DOFat that grid point, defined in the E system, by the relationship:

$$\begin{aligned} \boldsymbol{U}_c^i &= \boldsymbol{T}_{CE}^i \; \boldsymbol{U}_E^i \\ \theta_c^i &= \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \boldsymbol{T}_{CE}^i \; \theta_E^i \end{aligned} \tag{A-10}$$

The extra transformation in the rotational case is a result of the difference in the definition of rotations for C and E systems (Figures A-3 and A-4).

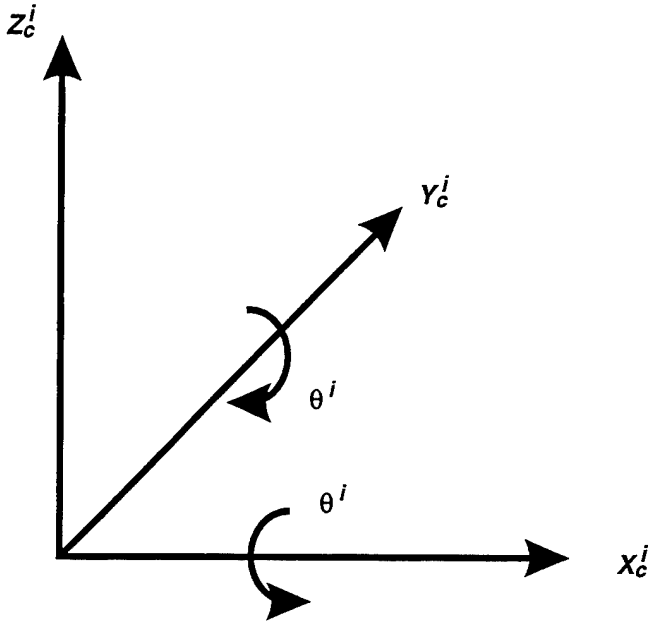


Figure A-4. Deformations in the Global Direction

The same five DOF are related to six dof's in the *I* system by using the transformations developed in Equations A-7 and A-8. Considering Equation A-10 and A-3:

$$\boldsymbol{U}_I = \boldsymbol{T}_{IE} \sum_{i=1}^{4} N_i \left[\boldsymbol{T}_{CE}^i\right]^T \boldsymbol{U}_c^i = \boldsymbol{T}_{IE} \sum_{i=1}^{4} N_i \boldsymbol{U}_E^i = \sum_{i=1}^{4} N_i \boldsymbol{T} \boldsymbol{U}_E^i \tag{A-11}$$

and

$$\begin{aligned} \theta_I &= \boldsymbol{T}_{IE} \sum_{i=1}^{4} N_i \left[\boldsymbol{T}_{CE}^i\right]^T \theta_c^i \\ &= \boldsymbol{T}_{IE} \sum_{i=1}^{4} N_i \left[\boldsymbol{T}_{CE}^i\right]^T \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \boldsymbol{T}_{CE}^i \theta_E^i \\ &= \sum_{i=1}^{4} N_i \boldsymbol{A}^i \theta_E^i \end{aligned} \tag{A-12}$$

Note that while $\boldsymbol{T}$ is invariant, $\boldsymbol{A}$ depends on the direction of the normal to mid-surface at each grid point.

At a point along the Z-axis of I system, at a level of $Z = \frac{\zeta}{2} t_I$, where,

$$t_I = \sum_{i=1}^{4} N_i t_i$$

is the thickness of the element evaluated at this particular integration point, the DOF in I system may be written in the following form:

$$\boldsymbol{u}_{MI} = \boldsymbol{u}_I \;\; ; \;\; \boldsymbol{u}_{BI} = \frac{\zeta}{2} t_I \theta_I \tag{A-13}$$

The strain-displacement relationships can now be developed, using these rearranged dof's:

$$\varepsilon_{MI} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix}_I^M = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix}_I^M = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{bmatrix} \boldsymbol{u}_{MI} \tag{A-14}$$

$$\varepsilon_{BI} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix}_I^B = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix}_I^B = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{bmatrix} \boldsymbol{u}_{BI} \tag{A-15}$$

$$\gamma_{sI} = \begin{Bmatrix} \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix}_I = \begin{Bmatrix} \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \end{Bmatrix} = \begin{bmatrix} 0 & 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} & 0 \\ 0 & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial z} & 0 & 0 \end{bmatrix} \begin{Bmatrix} u_M \\ u_B \end{Bmatrix}_I \tag{A-16}$$

Inserting Equations A-11 through A-13 into Equations A-14 through A-16, and considering the following:

$$\frac{\partial}{\partial z}\, \boldsymbol{u}_{BI} = \frac{\partial}{\partial z}\, z\, \theta_I = \theta_I$$

$$\begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ 1 \end{Bmatrix} = \sum_{i=1}^{4} \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ N_i \end{Bmatrix} \tag{A-17}$$

we arrive at the following general relationships:

$$\varepsilon_{MI} = \sum_{i=1}^{4} \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_i}{\partial y} & 0 \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 \end{bmatrix} \boldsymbol{T}\, \boldsymbol{U}_E^i \tag{A-18}$$

$$\varepsilon_{BI} = \frac{\zeta}{2} t_I \sum_{i=1}^{4} \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_i}{\partial y} & 0 \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 \end{bmatrix} \boldsymbol{A}^i\, \theta_E^i \tag{A-19}$$

$$\gamma_{sI} = \sum_{i=1}^{4} \begin{bmatrix} 0 & 0 & \frac{\partial N_i}{\partial y} & 0 & N_i & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial x} & N_i & 0 & 0 \end{bmatrix} \begin{Bmatrix} \boldsymbol{T} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}^i \end{Bmatrix} \begin{Bmatrix} \boldsymbol{U} \\ \theta \end{Bmatrix}_E^i \tag{A-20}$$

or, collectively:

$$\varepsilon_I = \begin{Bmatrix} \varepsilon_M \\ \varepsilon_B \\ \gamma_s \end{Bmatrix}_I \sum_{i=1}^{4} \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial N_i}{\partial y} & 0 & 0 & 0 & 0 \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\zeta t_i}{2}\frac{\partial N_i}{\partial x} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\zeta t_i}{2}\frac{\partial N_i}{\partial y} & 0 \\ 0 & 0 & 0 & \frac{\zeta t_i}{2}\frac{\partial N_i}{\partial y} & \frac{\zeta t_i}{2}\frac{\partial N_i}{\partial x} & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial y} & 0 & N_i & 0 \\ 0 & 0 & \frac{\partial N_i}{\partial x} & N_i & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{T} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}^i \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \\ \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix}_E^i \tag{A-21}$$

Since the shape functions N_i are defined in terms of the curvilinear coordinates (ξ, η), the shape function derivatives are related to the corresponding Cartesian derivatives in the element $\boldsymbol{E}$ coordinate system, by using the rules of partial differentiation, as:

$$\begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} \tag{A-22}$$

The first and second rows of the transformation matrix (or Jacobian matrix $\boldsymbol{J}$) are the tangent vectors to the surface τ = constant and the third row is the interpolated values of the nodal normals. (Note the nodal normals are evaluated by carrying out the cross product of the two tangent vectors at the node point.)

From Equation A-7 the coordinates in the "I" system are related to the coordinates in the "E" system by the following:

$$\boldsymbol{U}_I = \boldsymbol{T}_{IE}\ \boldsymbol{U}_E$$

Therefore, the derivatives are given by:

$$\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} = \Phi \begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{Bmatrix} \tag{A-23}$$

$$\Phi = \boldsymbol{T}_{IE}\, \boldsymbol{J}^{-1} = \begin{bmatrix} \varphi_{11} & \varphi_{12} & 0 \\ \varphi_{21} & \varphi_{22} & 0 \\ \varphi_{31} & \varphi_{32} & \varphi_{33} \end{bmatrix}$$

Note that $\frac{\partial N_i}{\partial \zeta}$ and $\frac{\partial N_i}{\partial z}$ will be zero when the interpolated normal at the integration point coincides with the normal to the mid-surface; e.g., in the case of the flat plate (φ_{31} and φ_{32} are zero). The zero terms in φ, i.e., θ_{13} and θ_{23}, result from dot products of perpendicular vectors.

A.3. STRESS-STRAIN RELATIONSHIPS

Stresses are related to the previously defined strains by the elasticity matrix $\boldsymbol{G}$ (where $\boldsymbol{G}$ is partitioned to give separate membrane stresses).

$$\begin{Bmatrix} \sigma_M \\ \sigma_B \\ \tau_{TS} \end{Bmatrix} = \begin{bmatrix} \boldsymbol{G}_1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_2 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{G}_3 \end{bmatrix} \begin{Bmatrix} \varepsilon_M \\ \varepsilon_B \\ \gamma_{TS} \end{Bmatrix}_{MEC} - \begin{Bmatrix} \varepsilon_M \\ \varepsilon_B \\ \boldsymbol{0} \end{Bmatrix}_T \tag{A-24}$$

$$\sigma_I = \boldsymbol{G}_I \left(\varepsilon_{MEC} - \varepsilon_T \right)_I$$

where

σ_M	Membrane stress vector
σ_B	Bending stress vector
τ_{TS}	Transverse shear stress vector
$\boldsymbol{G}_1$	Membrane moduli matrix
$\boldsymbol{G}_2$	Bending moduli matrix
$\boldsymbol{G}_3$	Transverse shear moduli matrix

and subscripts "MEC" and "T" refer to mechanical and thermal, respectively.

The membrane-bending coupling moduli matrix G_4 will be incorporated into the G matrix following this discussion of the uncoupled matrices.

All anisotropic, orthotropic and isotropic material properties are supported. The elastic modulus matrix G_M is defined in the material coordinate system and transformed into the user defined element coordinate system by means of a transformation angle, θ_M, which references the user defined element **X-AXIS** or the material coordinate system ID (MCSID) specified by the user. θ_M is in the X-Y plane of the element as shown in Figure A-5.

The elastic modulus matrix in the element coordinate system is:

$$G_I = U^T \, G_M \, U \tag{A-25}$$

(Note that since the projection of X_I onto the $X_E - Y_E$ plane is parallel to X_E, no extra transformations are required between the "E" and "I" systems.)

The transformation matrix for G_1, G_2 and G_4 is:

$$U_1 = \begin{bmatrix} \cos^2\theta_M & \sin^2\theta_M & \cos\theta_M \sin\theta_M \\ \sin^2\theta_M & \cos^2\theta_M & -\cos\theta_M \sin\theta_M \\ -2\sin\theta_M \cos\theta_M & 2\sin\theta_M \cos\theta_M & \cos^2\theta_M - \sin^2\theta_M \end{bmatrix} \tag{A-26}$$

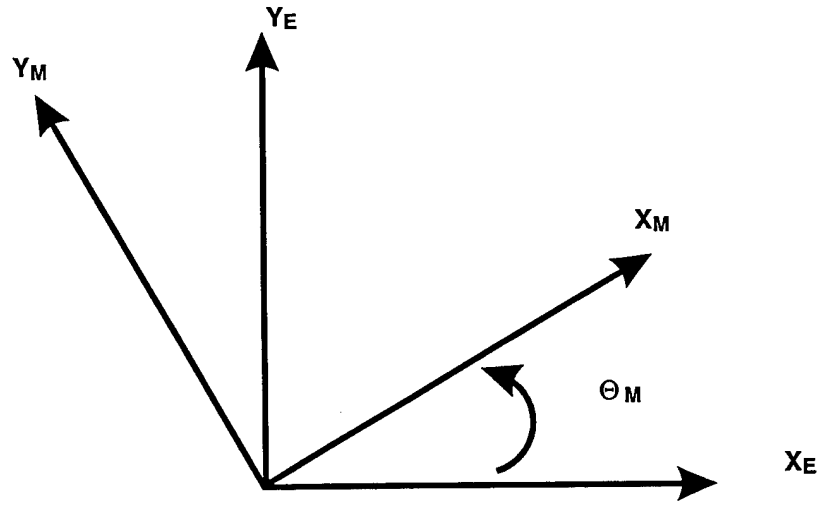


Figure A-5. Material and User Defined Element Axes

and the transformation matrix for G_3 is:

$$U_2 = \begin{bmatrix} \cos\theta_M & \sin\theta_M \\ -sin\theta_M & \cos\theta_M \end{bmatrix} \tag{A-27}$$

For isotropic materials:

A. Membrane

$$G_1 = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ & \nu & 0 \\ SYM & & \frac{1-\nu}{2} \end{bmatrix} \tag{A-28}$$

B. Bending

$$G_2 = \frac{t^3}{12I} G_1 \tag{A-29}$$

C. Transverse Shear

$$G_3 = \frac{t_s}{t} \begin{bmatrix} \beta_1 \frac{(1-\nu)}{2K} & 0 \\ 0 & \beta_2 \frac{(1-\nu)}{2K} \end{bmatrix} \tag{A-30}$$

where E is the Young's modulus; t is the element thickness at the corresponding integration point, ν is the Poisson's ratio and $\frac{t_s}{t}$ is the transverse shear factor.

Note that in matrix G_3, the factor "K" is introduced to compensate for the difference in shear distribution though the thickness, which is parabolic and not constant as indicated by the displacement function. The value of $K = 1.2$ is the ratio of the relevant strain energies. The β_i factors, which are derived numerically, are introduced to compensate for the "locking" of the element due to excessive shear stiffness.

For anisotropic materials:

A. Membrane

$$G_1 = \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ & G_{22} & G_{23} \\ SYM & & G_{33} \end{bmatrix} \tag{A-31}$$

B. Bending

$$G_2 = \frac{t^3}{12I} G_1 \tag{A-32}$$

C. Transverse Shear

$$G_3 = \frac{t_s}{t} \begin{bmatrix} G_{11} & G_{12} \\ G_{12} & G_{22} \end{bmatrix} \tag{A-33}$$

For orthotropic materials:

A. Membrane

$$G_1 = \frac{1}{1 - \nu_{12}\nu_{21}} \begin{bmatrix} E_1 & \nu_{12}E_2 & 0 \\ & E_2 & 0 \\ SYM & & G_{12}(1 - \nu_{12}\nu_{21}) \end{bmatrix} \tag{A-34}$$

B. Bending

$$G_2 = \frac{t^3}{12I} G_1 \tag{A-35}$$

C. Transverse shear

$$G_3 = \frac{t_s}{t} \begin{bmatrix} G_{1z} & 0 \\ 0 & G_{2z} \end{bmatrix} \tag{A-36}$$

where E_1 and E_2 are the Young's moduli in the principal material axes, ν_{12} is the major Poisson's ration; G_{12} is the in-plane shear modulus, G_{1z} and G_{2z} are the out-of-plane shear moduli and $\frac{t_s}{t}$ is the transverse shear factor.

The derivation of the G_4 membrane-bending coupling matrix begins by denoting the strains at the mid-surface as:

$$\varepsilon_M^o = \begin{Bmatrix} \varepsilon_x^o \\ \varepsilon_y^o \\ \gamma_{xy}^o \end{Bmatrix} \tag{A-37}$$

and the out of plane curvatures as:

$$K = \begin{Bmatrix} K_x \\ K_y \\ K_{xy} \end{Bmatrix} \tag{A-38}$$

Therefore, the strains at a distance z above the mid-surface of the element are:

$$\varepsilon = \varepsilon_M^o - z\,\boldsymbol{K} \tag{A-39}$$

The corresponding 2-D stresses are:

$$\sigma = \boldsymbol{G}_I \left[\varepsilon_M^o - z\,\boldsymbol{K} \right] \tag{A-40}$$

where $\boldsymbol{G}_I$ is a (3x3) matrix of elastic moduli.

The forces and moments per unit length are therefore given by:

$$\boldsymbol{F} = \int_{z_a}^{z_b} \sigma\, dz = \int_{z_a}^{z_b} \boldsymbol{G}_I \left[\varepsilon^o - z\boldsymbol{K} \right] dz \tag{A-41}$$

$$\boldsymbol{F} = t\,\boldsymbol{G}_I\, \varepsilon^o + t^2\, \boldsymbol{G}_4 \boldsymbol{K}$$

$$\boldsymbol{M} = \int_{z_a}^{z_b} \sigma\, z dz = \int_{z_a}^{z_b} \boldsymbol{G}_I \left[-z\,\varepsilon^o + z^2 \boldsymbol{K} \right] dz \tag{A-42}$$

$$\boldsymbol{M} = t^2\, \boldsymbol{G}_4\, \varepsilon^o + I\, \boldsymbol{G}_2 \boldsymbol{K}$$

where t is the plate thickness and I is the bending inertia. Assuming a linear variation of elastic properties between top and bottom surface.

$$\boldsymbol{G}_1 = \frac{1}{t} \int_{-\frac{t}{2}}^{\frac{t}{2}} G dz = \frac{G_t + G_B}{2} \tag{A-43}$$

$$\boldsymbol{G}_2 = \frac{1}{I} \int_{-\frac{t}{2}}^{\frac{t}{2}} G dz = \frac{t^3}{12I}\, \boldsymbol{G}_1 \tag{A-44}$$

$$\boldsymbol{G}_4 = \frac{1}{t^3} \int_{-\frac{t}{2}}^{\frac{t}{2}} (-z)\, G dz = -\left(\frac{G_t - G_B}{12} \right) \tag{A-45}$$

Notethatthemembrane-bendingstiffnesscouplingtermsvanishforanelementwhoseelasticproperties aresymmetricrelativetothemeanplaneoftheelement.

By assuming that the elastic modulus has a linear variation between the top and bottom surfaces, define:

$$G = G_1 + \frac{\zeta}{2\left(G_T - G_B\right)} \tag{A-46}$$

Therefore, from Equations A-31 and A-32:

A. Membrane

$$G = G_1 + \frac{\zeta}{2}\left[-12G_4\right] \tag{A-47}$$

$$G = G_1 - 6\zeta G_4 \tag{A-48}$$

B. Bending

$$G_2 = \frac{G_1 t^3}{12I}$$

$$G = \frac{12I}{t^3} G_2 - 6\zeta G_4 \tag{A-49}$$

Matrix G_3 is not affected since transverse shears are assumed to have no coupling action.

Therefore, the stress-strain relationship, allowing for membrane, bending, transverse shear and membrane-bending coupling is:

$$\begin{Bmatrix} \sigma_M \\ \sigma_{TOT} \\ \tau_{TS} \end{Bmatrix} = \begin{Bmatrix} G_1 & G_1 - 6\zeta G_4 & 0 \\ G_1 - 6\zeta G_4 & G_2 & 0 \\ 0 & 0 & G_3 \end{Bmatrix} \begin{Bmatrix} \varepsilon_M \\ \varepsilon_B \\ \gamma_{TS} \end{Bmatrix} \tag{A-50}$$

where

$$\sigma_M = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} \qquad \text{Membrane stresses}$$

$$\sigma_{TOT} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}_{TOT} \qquad \text{Total membrane and bending stresses}$$

$$\sigma_{TS} = \begin{Bmatrix} \tau_{yz} \\ \\ \tau_{xz} \end{Bmatrix} \qquad \text{Transverse shear stresses}$$

$$\varepsilon_M = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix}$$ Membrane strain

$$\varepsilon_B = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix}_B$$ Bending strains

$$\gamma_{TS} = \begin{Bmatrix} \gamma_{yz} \\ \\ \gamma_{xz} \end{Bmatrix}$$ Transverse shear strains

A.4. STIFFNESS MATRIX

The element stiffness matrix is derived by minimizing the total potential energy and is given in numerical form by employing the Gauss-quadrature integration method:

$$\boldsymbol{K}_E = \sum \sum \sum \boldsymbol{B}^T \boldsymbol{G} \boldsymbol{B} W_\xi W_\eta W_\zeta \det \boldsymbol{J} \tag{A-51}$$

where (ξ, η, ζ) are the Gaussian integration point coordinates and W_ξ, W_η, and W_ζ are the associated weight factors. Det $\boldsymbol{J}$ represents the physical volume of the element as calculated at this point, B is the strain displacement relationship of Equation A-21 and G is the stress strain relationship of Equation A-50.

Each element stiffness matrix partition in the element coordinate system, $\boldsymbol{K}_{ijEE}$, is transformed to the global coordinate system by the following transformation:

$$\boldsymbol{K}_{ijG} = \boldsymbol{T}_{EG_i}^T \boldsymbol{K}_{ijEE} \boldsymbol{T}_{EG_i} \tag{A-52}$$

where $\boldsymbol{T}_{EG_i}$ is determined by relating the element coordinate system to the global coordinate system for grid i through the basic coordinate system:

$$\boldsymbol{T}_{EG_i} = \boldsymbol{T}_{EB_i} \boldsymbol{T}_{BG_i} \tag{A-53}$$

A.5. CONSISTENT AND LUMPED MASS MATRICES

The consistent mass matrix terms are evaluated, neglecting the rotational inertias associated with the α and β degrees of freedom, by the following expression:

$$M_{ij} = \sum_{n=1}^{4} N_i N_j \rho \ |J| \ t_n \tag{A-54}$$

where N_i is the shape function for node i, ρ is the mass per unit volume, $|J|$ is the physical area of the element and t_n is the element thickness at the integration point.

The lumped mass matrix, which is calculated at the pseudo center (i.e., the average of the element grid coordinates), is prorated to the edges based on the distance of the pseudo center from each edge.

The terms of the lumped mass matrix are evaluated using:

$$M_{ij} = \sum_{n=1}^{4} N_i \; \rho \; |J| \; t_n \tag{A-55}$$

The transformation of the mass matrix to the global coordinate system is carried out using the same transformation matrices as used for the stiffness matrix in Equation A-52.

A.6. STRESS RECOVERY

The element stresses in partitioned form from Equation A-50 are

$$\begin{Bmatrix} \sigma_M \\ \sigma_{TOT} \\ \tau_{TS} \end{Bmatrix} = \begin{bmatrix} G_1 & G_1 - 6\zeta G_4 & 0 \\ G_1 - 6\zeta G_4 & G_2 & 0 \\ 0 & 0 & G_3 \end{bmatrix} \left\{ \begin{Bmatrix} \varepsilon_M \\ \varepsilon_B \\ \gamma_{TS} \end{Bmatrix}_{MEC} - \begin{Bmatrix} \varepsilon_M \\ \varepsilon_B \\ \gamma_{TS} \end{Bmatrix} \right\}$$

$$\begin{Bmatrix} \sigma_M \\ \sigma_{TOT} \\ \tau_{TS} \end{Bmatrix} = \begin{bmatrix} G_1 & G_4' & 0 \\ G_4' & G_2 & 0 \\ 0 & 0 & G_3 \end{bmatrix} \left\{ \begin{Bmatrix} \varepsilon_M \\ \varepsilon_B \\ \gamma_{TS} \end{Bmatrix}_{MEC} - \begin{Bmatrix} \varepsilon_M \\ \varepsilon_B \\ \gamma_{TS} \end{Bmatrix}_{MEC} \right\} \tag{A-56}$$

For a specified grid point temperature, the thermal strain vector is:

$$\varepsilon_{MT} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix}_T = \alpha_I \left(T_i - T_o \right) \tag{A-57}$$

where $\alpha_I = \boldsymbol{U}^{-1} \alpha_M$ is a vector of thermal expansion coefficients in the element coordinate system. $\boldsymbol{U}$ is the strain transformation matrix given in Equation A-26 and α_M is the vector of thermal expansion coefficients in the material axes. T_i and T_o are the specified grid point temperature and mid-surface (stress-free) temperature, respectively.

For a thermal gradient T', the thermal strain vector ε_{BT} is:

$$\varepsilon_{BT} = \alpha_I \left(\frac{\zeta t}{2} T' \right) \tag{A-58}$$

For thermal moments M_T, the thermal strain vector ε_{BT} is:

$$\varepsilon_{BT} = \frac{-\zeta t}{2I} G_2' M_T \tag{-A59}$$

NOTE: ASTROS does not support thermal gradient or moments so that the above equations are provided for completeness only.

The in-plane stress vector σ_z at fiber distance z from the mid-surface is:

$$\sigma_z = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}_z = \left(\frac{1}{2} - \frac{z}{t} \right) \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}_1 + \left(\frac{1}{2} + \frac{z}{t} \right) \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}_1 \tag{A-60}$$

where the stress vectors $\sigma_x, \sigma_y, \tau_{xy}\ 1^T$ and $\sigma_x, \sigma_y, \tau_{xy}\ 2^T$ are the bottom and top fiber stress vectors, respectively.

If a temperature T_i is specified at the point where outer fiber stresses are to be calculated, the additional thermal stress due to the difference between the specified temperature and a temperature that would be produced by a uniform thermal gradient T' or thermal moments M_T is calculated using:

$$\Delta \sigma_T = G_2\ \alpha_I \left(T_i - T_o - T' z \right) \tag{A-61}$$

for a thermal gradient T', and

$$\Delta \sigma_t = -z \frac{M_T}{I} - G_2\ \alpha T_i \tag{A-62}$$

A.7. FORCE RESULTANTS

The forces at the mid-surface are evaluated by taking the average stress values over the element thickness:

A. Forces

$$F = \begin{Bmatrix} F_x \\ F_y \\ F_{xy} \end{Bmatrix} = \left(\sigma_{z1} + \sigma_{z2} \right) \frac{t}{2} \tag{A-63}$$

B. Moments

$$\boldsymbol{M} = \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = \left(\sigma_{z1} + \sigma_{z2}\right)\frac{I}{2} \tag{A-64}$$

C. Transverse Shear Forces

$$\boldsymbol{Q} = \begin{Bmatrix} Q_x \\ \\ Q_y \end{Bmatrix} = \left(\tau_{z1} + \tau_{z2}\right)\frac{t}{2} \tag{A-65}$$

where stress vectors σ_{z1}, σ_{z2} are stresses at the integration points (default option) or at grid points (if requested) and, similarly, τ_{z1} and τ_{z2} are the transverse shear stresses.

A.8. THERMAL LOAD VECTOR

The thermal load vector is computed as:

$$\boldsymbol{P}_T = \int_v \boldsymbol{B}\ \boldsymbol{G}\ \varepsilon_T\ dv \tag{A-66}$$

where the load vector $\boldsymbol{P}_T$ is defined as:

$$\boldsymbol{P}_T = \begin{Bmatrix} \boldsymbol{F}_T \\ \boldsymbol{M}_T \end{Bmatrix} \tag{A-67}$$

where $\boldsymbol{F}_T$ and $\boldsymbol{M}_T$ are the thermal forces and moments, respectively.

The thermal strain vector is:

$$\varepsilon_T = \begin{Bmatrix} \varepsilon_{M_T} \\ \varepsilon_{B_T} \end{Bmatrix} = \begin{Bmatrix} \alpha_M \\ \alpha_B \end{Bmatrix} \Delta T \tag{A-68}$$

where ε_{M_T} and ε_{B_T} are the thermal membrane and bending strains, and correspondingly α_M and α_B are the thermal coefficients of expansion for membrane and bending. ΔT is dependent on the temperature loading being specified.

A. For a specified grid point temperature the thermal membrane strain vector, ε_{M_T}, is:

$$\varepsilon_M = \alpha_M \left(T_i - T_o\right) \tag{A-69}$$

T_i = Grid point temperature

T_o = Reference (stress-free) temperature

B. For a thermal gradient, the thermal bending strain vector, ε_{B_T}, is:

$$\varepsilon_B = \alpha_B \left(-\frac{\zeta t}{2} T' \right) \tag{A-70}$$

C. For thermal moments, the thermal bending strain vector, ε_{B_T}, is:

$$\varepsilon_B = G_2 \, M_T \, \frac{\tau t}{2I} \tag{A-71}$$

NOTE: ASTROS does not support thermal gradients or moments so that the above equations are provided for completeness only.

A.9. LAMINATED COMPOSITE MATERIALS

The capability to model a stack of layers with a single QUAD4 element is detailed including the computation of equivalent "single layer" properties, i.e., membrane, bending transverse shear and membrane-bending coupling. The recovery of element forces, layer and interlaminar shear stresses and the computation of ply failure indices is also described in the following overview of theory.

A.9.1. Overview of Theory

The calculation of the "overall" properties for the laminated composite elements is based on the classical lamination theory with the following assumptions:

A. Each of the lamina is in a state of plane stress.

B. The laminate is presumed to consist of perfectly bonded lamina.

C. The bonds are presumed to be infinitesimally thin and non-shear deformable. That is, the displacements are continuous across the lamina boundaries so that lamina can not slip relative to one another. Thus, the laminate behaves as a single layer with "special" properties.

The material properties of laminated composite materials are reflected in the following force-strain relationship:

$$\begin{Bmatrix} \boldsymbol{F} \\ \boldsymbol{M} \\ \boldsymbol{V} \end{Bmatrix} = \begin{bmatrix} t\,\boldsymbol{G}_1 & t^2\,\boldsymbol{G}_4 & \mathbf{0} \\ t^2\,\boldsymbol{G}_4 & I\,\boldsymbol{G}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & t_s\,\boldsymbol{G}_3 \end{bmatrix} \begin{Bmatrix} \varepsilon_M - \varepsilon_M^T \\ \boldsymbol{K} - \boldsymbol{K}^T \\ \gamma \end{Bmatrix} \tag{A-72}$$

where

$$\boldsymbol{F} = \begin{Bmatrix} F_x \\ F_y \\ F_{xy} \end{Bmatrix}$$ Membrane forces per unit length.

$$\boldsymbol{M} = \begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix}$$ Bending moments per unit length.

$$\boldsymbol{V} = \begin{Bmatrix} V_x \\ \\ V_y \end{Bmatrix}$$ Transverse shear forces per unit length.

and the remaining terms have been defined previously.

The $\boldsymbol{G}_1$, $\boldsymbol{G}_2$, and $\boldsymbol{G}_4$ terms are defined by the following:

$$\boldsymbol{G}_1 = \frac{1}{t}\int \boldsymbol{G}_E \, dz$$

$$\boldsymbol{G}_2 = \frac{1}{I}\int z^2 \, \boldsymbol{G}_E \, dz \qquad \text{(A-73)}$$

$$\boldsymbol{G}_4 = \frac{1}{t}\int -z \, \boldsymbol{G}_E \, dz$$

The limits on the integration are from the bottom surface to the top surface of the laminated composite. The elasticity matrix $\boldsymbol{G}_E$ has the following form for isotropic materials:

$$\boldsymbol{G}_E = \begin{bmatrix} \frac{E}{1-\nu^2} & \nu\frac{E}{1-\nu^2} & 0 \\ SYM & \frac{E}{1-\nu^2} & 0 \\ & & 0 \end{bmatrix} \qquad \text{(A-74)}$$

$$G = \frac{E}{2(1+\nu)} \qquad \text{(A-75)}$$

For orthotropic materials, matrix [GE] is:

$$\boldsymbol{G}_E = \begin{bmatrix} \frac{E_1}{1-\nu_1\nu_2} & \frac{\nu_1 E_2}{1-\nu_1\nu_2} & 0 \\ SYM & \frac{E_2}{1-\nu_1\nu_2} & 0 \\ & & G_{12} \end{bmatrix} \qquad \text{(A-76)}$$

Equation A-73 may be rewritten as:

$$G_{ij_1} = \frac{1}{t} \sum_{k=1}^{N} \overline{G}_{ij}^{k} \left(Z_k - Z_{k-1} \right)$$

$$G_{ij_2} = \frac{1}{3I} \sum_{k=1}^{N} \overline{G}_{ij}^{k} \left(Z_k^3 - Z_{k-1}^3 \right) \qquad \text{(A-77)}$$

$$G_{ij} = -\frac{1}{2t^2} \sum_{k=1}^{N} \overline{G}_{ij}^{k} \left(Z_k^2 - Z_{k-1}^2 \right)$$

where $\overline{G}_{ij}^{k}$ is the reduced moduli matrix evaluated for each lamina k after transforming the lamina property matrix from the fiber to the element material axes.

Z_k and Z_{k-1} are the top and bottom distances of lamina k from the geometric middle plane of the laminate, as illustrated in Figure A-6, and N is the number of laminae (or plies). Note that the plies are numbered serially starting with 1 at the bottom layer. The bottom layer is defined as the surface with the largest -z value in the element coordinate system. If the option to model membrane-only elements is exercised, matrices G_2, G_3, and G_4 are set to zero.

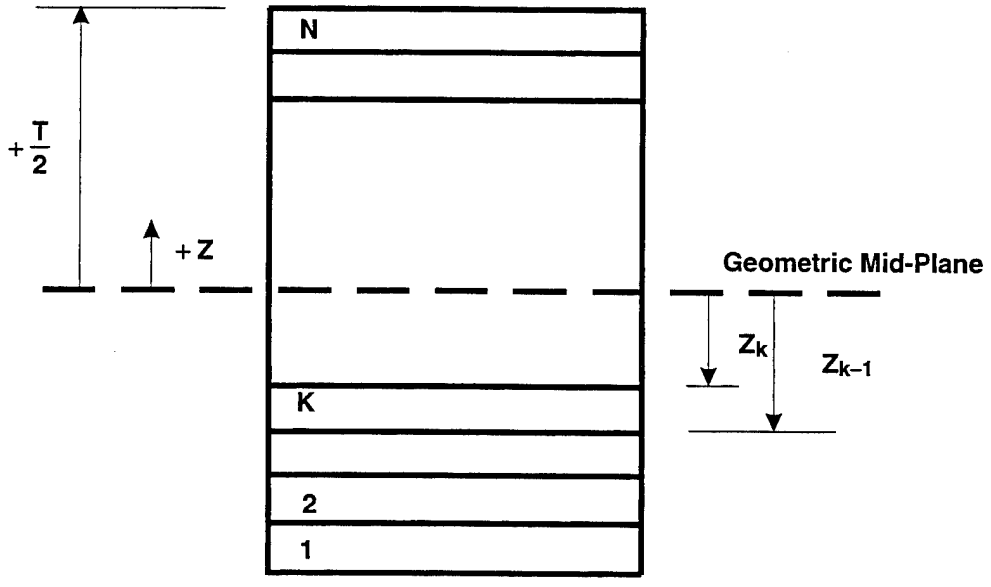


Figure A-6. Geometry of an N-Layered Element

If the user defined element axis is not coincident with the element material axis, the user specified transformation angle θ_M, which references the element X-axis, is added to the layer orientation angle. The property matrices $\boldsymbol{G}_1$, $\boldsymbol{G}_2$, and $\boldsymbol{G}_4$ are then transformed to the user defined element axis using the following equation:

$$\boldsymbol{G}_E = \boldsymbol{U}^T \boldsymbol{G}_M \boldsymbol{U} \tag{A-78}$$

where

$$\boldsymbol{U} = \begin{bmatrix} \cos^2\theta & \sin^2\theta & \cos\theta \sin\theta \\ \sin^2\theta & \cos^2\theta & -\cos\theta \sin\theta \\ -2\sin\theta\cos\theta & 2\sin\theta\cos\theta & \cos^2\theta - \sin^2\theta \end{bmatrix} \tag{A-79}$$

The transverse shear flexibility $\boldsymbol{G}_3$, matrix is defined by:

$$\boldsymbol{G}_3 = \begin{bmatrix} G_{11} & G_{12} \\ G_{12} & G_{22} \end{bmatrix} \tag{A-80}$$

and the corresponding matrix transformed into the user-defined element coordinate system is given by:

$$\boldsymbol{G} = \boldsymbol{W}^T \boldsymbol{G}_M \boldsymbol{W} \tag{A-81}$$

where

$$\boldsymbol{W} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{A-82}$$

The derivation of the transverse shear flexibility matrix $\boldsymbol{G}_3$ for the laminate is considered next.

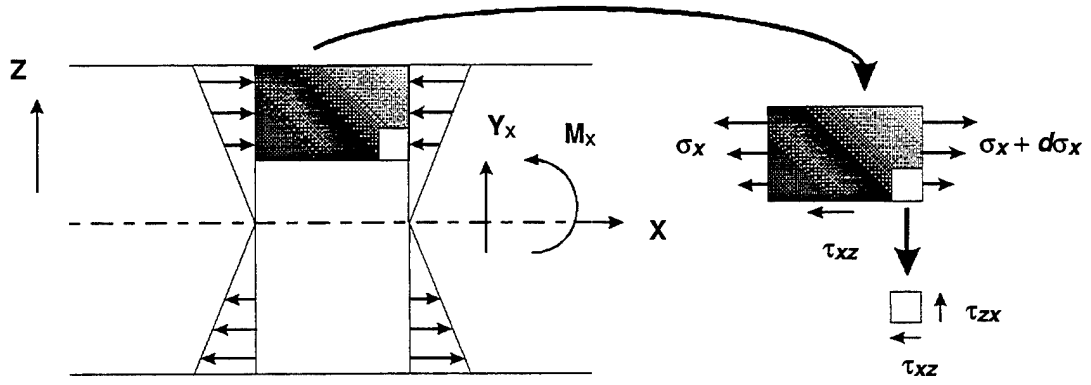
The mean value of the transverse shear modulus, $\boldsymbol{G}$, for the laminated composite is defined in terms of the transverse shear strain energy, $\boldsymbol{U}$, through the depth as:

$$U = \frac{V^2}{2Gt} = \frac{1}{2} \int \frac{\tau^2(z)}{G(z)}\, dz \tag{A-83}$$

A unique mean value of transverse shear strain is assumed to exist for both the x- and y-components of the element coordinate system, but for ease of discussion, only the evaluation of an uncoupled x-component of the shear moduli will be illustrated here. From Equation A-83, the mean value of transverse shear modulus is written in the following form:

$$\frac{1}{G_x} = \frac{t}{V_x^2} \sum_{i=1}^{N} \int_{Z_{i-1}}^{Z_i} \frac{\left(\tau_{zx}(x)\right)^2}{(G_x)_i}\, dz \tag{A-84}$$

where G is an "average" transverse shear coefficient used by the element code and $(G_x)_i$ is the local shear coefficient for layer *i*. To evaluate Equation A-84, it is necessary to obtain an expression for $\tau_{zx}(z)$. This is accomplished by assuming that the x- and y-components of stress are decoupled from one another. This assumption allows the desired equation to be deduced through an examination of a beam of unit cross-sectional width.



The equilibrium conditions in the horizontal direction and for total moment are:

$$\frac{\partial \tau_{xz}}{\partial z} - \frac{\partial \sigma_x}{\partial x} = 0 \qquad \text{(A-85)}$$

$$V_x + \frac{\partial M_x}{\partial x} = 0 \qquad \text{(A-86)}$$

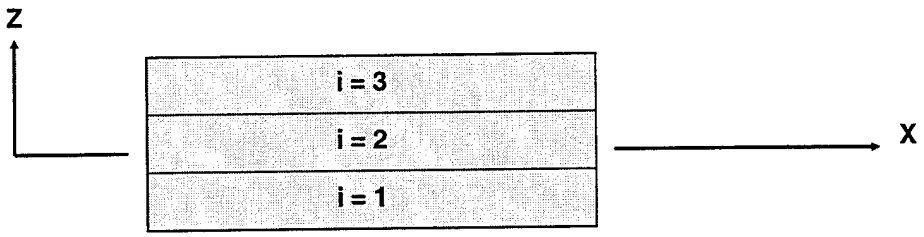
If the location of the neutral surface is denoted by z_x and ρ is the radius of curvature of the beam, the axial stress, σ_x, is expressed in the form:

$$\sigma_x = \frac{E_x\left(\bar{z}_x - z\right)}{(EI)_x} M_x \qquad \text{(A-87)}$$

Equation A-87 is differentiated with respect to x and combined with Equations A-85 and A-86. For constant E_x, the result is integrated to yield the following expression:

$$\tau_{xz} = C_i + \frac{V_x}{(EI)_x}\left(\bar{z}_x z - \frac{z^2}{2}\right)E_{x_i} \qquad z_{i-1} < z < z_i \qquad \text{(A-88)}$$

Equation A-88 is used in the analysis of n-ply laminates because sufficient conditions exist to determine the constants C_i $(i = 1, 2, ... n)$ and the "directional bending center," z_x. For example, consider the following laminated configuration:



At the bottom surface ($i = 1, z = z_{o,}$ and $\tau_{xz} = 0$), therefore:

$$C_1 + \frac{-V_x}{(EI)_x}\left(\bar{z}_x z_o - \frac{z_o^2}{2}\right)E_{x_1} \tag{A-89}$$

and for the first ply at the interface between plies $i = 1$ and $i = 2$ $\left(z = z_1\right)$:

$$\left(\tau_{xz}\right)_1 = \frac{V_x}{(EI)_x}\left[z_x\left(z_1 - z_0\right) - \frac{1}{2}\left(z_1^2 - z_0^2\right)\right]E_{x_1} \tag{A-90a}$$

At this interface between plies $i = 1$ and $i = 2$:

$$\left(\tau_{xz}\right)_2 = C_2 + \frac{V_x}{(EI)_x}\left(\bar{z}_x z_1 - \frac{z_1^2}{2}\right)E_{x_2} \tag{A-90b}$$

and since $\left(\tau_{xz}\right)_2 = \left(\tau_{xz}\right)_1$ at $z = z_1$:

$$C_2 = \left(\tau_{xz}\right)_1 - \frac{V_x E_{x2}}{(EI)_x}\left(\bar{z}_x z_1 - \frac{1}{2} z_1^2\right) \tag{A-91}$$

Then, in the ply, $z_1 < z < z_2$, the shear is:

$$\tau_{xz^{(z)}} = \left(\tau_{xz}\right)_1 + \frac{V_x E_{x2}}{(\overline{EI})_x}\left[\bar{z}_x\left(z - z_1\right) - \frac{1}{2}\left(z^2 - z_1^2\right)\right] \tag{A-92}$$

In general, for any ply $z_{i-1} < z < z_i$, the shear is:

$$\tau_{xz^{(z)_i}} = \left(\tau_{xz}\right)_{i-1} + \frac{V_x E_{xi}}{(\overline{EI})_x}\left[\bar{z}_x\left(z - z_{i-1}\right) - \frac{1}{2}\left(z^2 - z_{i-1}^2\right)\right] \tag{A-93}$$

At any ply interface, z_i, the shear is therefore:

$$\left(\tau_{xz}\right)_i = \frac{V_x}{(EI)_x}\sum_{j=1}^{i} E_{xj}\, T_j\left[\bar{z} - \frac{1}{2}\left(z_j + z_{j-1}\right)\right] \tag{A-94}$$

where $T_j = z_j - z_{j-1}$.

Note that the shear at the top face, $\left(\tau_{xz}\right)_n$, is zero and therefore:

$$\left(\tau_{xz}\right)_n = \frac{V_x}{(EI)_x}\left(\bar{z}_x \sum_{j=1}^{n} E_{xj}\, T_j \sum_{j=1}^{n} E_{xj}\, T_j \frac{\left(z_j + z_{j-1}\right)}{2}\right) = 0 \tag{A-95}$$

Equation A-95 proves that if z_x is the bending center, the shear at the top surface must be zero.

A better form of Equation A-93, for this purpose, is:

$$\tau_{xz}(z)_i = \frac{V_x E_{xi}}{(EI)_x}\left(fx_i + \bar{z}\left(z - z_{i-1}\right) - \frac{1}{2}\left(z^2 - z_{i-1}^2\right)\right) \tag{A-96}$$

where

$$fx_i = \frac{1}{E_{x_i}}\sum_{j-1}^{i-1} E_{x_j}\, T_j\left[\bar{z}_x - \frac{1}{2}\left(z_j + z_{j-1}\right)\right] \tag{A-97}$$

Substituting Equation A-96 into Equation A-84 yields:

$$\frac{1}{G_x} = \frac{T}{(EI)_x^2}\sum_{i=1}^{n} \frac{1}{G_{x_i}} R_{x_i} \tag{A-98a}$$

where

$$R_{x_i} = \left(E_{xi}\right)^2 T_i \left[\left(fx_i + \left(\bar{z}_x - z_{i-1} \right) T_i - \frac{1}{3} T_i^2 \right) fx_i + \left(\frac{1}{3} \left(\bar{z}_x - 2\, z_{i-1} \right) - \frac{1}{4} T_i \right) \bar{z}_x T_i^3 \right.$$
$$\left. + \left(\frac{1}{3} z_{i-1}^3 + \frac{1}{4} z_{i-1} T_i + \frac{1}{20} T_i^3 \right) T_i^3 \right] \qquad \text{(A-98b)}$$

This expression for the inverse shear modulus for the x-direction is generalized to provide for the calculation of each term in the two-by-two matrix of shear moduli as:

$$\bar{G}_{ki} = \left[\frac{T}{EI_{kk}^2} \sum_{i=1}^{n} G_{kl}^{i^{-1}} R_{ki} \right]^{-1} \qquad \text{(A-99)}$$

where

$k = 1, 2$
$l = 1, 2$

Note that if no shear is given, $G^{i^{-1}} = 0$, and also that, in Equation A-99:

$(\overline{EI})_{11}$ = 1, 1 term of $I \cdot G_2^*$

$(\overline{EI})_{22}$ = 2, 2 term of $I \cdot G_2^*$

where G_2^* is calculated in the same manner as G_3 except that Poisson's ratio is set to zero. The moduli for individual plies are provided through user input. Because $G_{12} \neq G_{21}$, in general, an average value is used for the coupling terms.

$$G_3 - \begin{bmatrix} G_{11} & \bar{G}_{12_{AVG}} \\ \bar{G}_{12_{AVG}} & G_{22} \end{bmatrix} \qquad \text{(A-100)}$$

A.9.2. Element Layer Stress Recovery

The linear strain variation is given by:

$$\varepsilon_x = \varepsilon_M - z\, K \qquad \text{(A-101)}$$

where

ε_x	Layer strain vector in the element coordinate system.
ε_M	Reference surface strain in the element coordinate system.
K	Reference surface curvatures in the element coordinate system.
z	Distance of the mid-surface of the layer k from the laminate reference surface.

The individual layer stress vector in the fiber coordinate system is:

$$\sigma_L = \boldsymbol{G}_L \; \boldsymbol{T} \; \varepsilon_x \tag{A-102}$$

where

σ_L Layer stress vector in the fiber coordinate system.

$\boldsymbol{G}_L$ Stress-strain matrix in the fiber coordinate system.

$\boldsymbol{T}$ Transformation matrix to transform strains from element coordinate system to fiber coordinate system.

ε_x Layer strain vector in the element coordinate system.

For element temperature and/or thermal gradients, the strain vector has to be corrected for thermal effects before applying Equation A-103:

$$\varepsilon_x = \varepsilon'_x - \alpha \; (T + zT') \tag{A-103}$$

and for thermal moments

$$\varepsilon_x = \varepsilon_x - \varepsilon_x^T \tag{A-104}$$

where

ε'_x Mechanical strains.

α Thermal coefficients of expansion in the element coordinate system.

T Element temperature.

T' Element thermal gradient.

z Distance from the middle of the layer to the laminate reference surface.

ε_x^T Layer strains due to thermal moments in the element coordinate system.

The thermal strain vector due to applied thermal moments is determined by substituting for $\boldsymbol{M}$ in Equation A-73 and solving for the reference surface strains and curvatures, $\varepsilon^T{}_M$ and $\boldsymbol{K}^T$, respectively.

A.9.3. Interlaminar Shear Stresses

The interlaminar shear stress τ_{yz}, τ_{xz} can be computed at any ply interface from Equation A-96.

A.9.4. Force Resultants

Forces and moments for the element are computed using:

$$\boldsymbol{F} = \sum_{i=1}^{N} \sigma_x \, T_i$$

$i = 1, \; N$ (number of layers) (A-105)

$$\boldsymbol{M} = \sum_{i=1}^{N} -z_i \, T_i \, \sigma_x$$

where

$\boldsymbol{F}$	In-plane force resultants.
$\boldsymbol{M}$	Out-of-plane moments.
σ_x	Stresses in the element coordinate system.
T_i	Layer thickness.
z_i	Distance from the middle of the layer to the laminate reference surface.

A.9.5. Failure Indices

Failure indices assume a value of one on the periphery of a failure surface in stress space. If the failure index is less than one, the lamina stress is interior to the periphery of the failure surface and the lamina is assumed "safe" and if it is greater than one the lamina is assumed to have "failed." The failure indices represent a phenomenological failure criterion, because only the occurrence of failure is predicted.

The analytical definition of a failure surface in stress space for a lamina subjected to biaxial (planar) states of stress is provided via the following failure theories.

1. HILL
2. HOFFMAN
3. TSAI-WU
4. MAXIMUM STRESS
5. MAXIMUM STRAIN

In the analysis of laminated composites, which are typically orthotropic materials (possibly exhibiting unequal properties in tension and compression), the strength of orthotropic lamina is a function of body orientation relative to the imposed stress. In order to determine the structural integrity of the lamina, a set of intrinsic strength properties (allowable stresses or allowable strains) in the principal material directions are defined as:

X_t	Ultimate uniaxial tensile strength in the fiber direction,
X_c	Ultimate uniaxial compressive strength in the fiber direction,
Y_t	Ultimate uniaxial tensile strength perpendicular to the fiber direction,
Y_c	Ultimate uniaxial compressive strength perpendicular to the fiber direction,
S	Ultimate planar shear strength under pure shear loading,

E_t Ultimate uniaxial tensile strain in the fiber direction,

E_c Ultimate uniaxial compressive strain in the fiber direction,

F_t Ultimate uniaxial tensile strain perpendicular to the fiber direction,

F_c Ultimate uniaxial compressive strain perpendicular to the fiber direction, and

E_s Ultimate planar shear strain under pure shear loading.

Each of these terms is an algebraically positive value. For most composite materials, the planar shear strengths and strains are equal for positive and negative shear loadings.

The five failure theories and a bonding failure index are now described:

HILL'S THEORY

$$\frac{\sigma_1^2}{X^2} + \frac{\sigma_2^2}{Y^2} - \frac{\sigma_1 \sigma_2}{X^2} + \frac{\tau_{12}^2}{S^2} = \text{FAILURE INDEX (FI)} \qquad \text{(A-106)}$$

and $X = X_t$ if σ_1 is positive, and X_c if σ_1 is negative; similarly for Y. For the interaction term, $\frac{\sigma_1 \sigma_2}{X^2}$, $X = X_t$ if the product $\sigma_1 \sigma_2$ is positive $X = X_c$ otherwise.

HOFFMAN'S THEORY

$$\left(\frac{1}{X_t} - \frac{1}{X_c}\right)\sigma_1 + \left(\frac{1}{Y_t} - \frac{1}{Y_c}\right)\sigma_2 + \frac{\sigma_1^2}{X_t X_c} + \frac{\sigma_2^2}{Y_t Y_c} + \frac{\tau_{12}^2}{S^2} + \frac{\sigma_1 \sigma_2}{X_t X_c} = \text{FI} \qquad \text{(A-107)}$$

Note that this theory takes into account the difference in the tensile and compressive allowable stresses by using linear terms in the failure equation.

TSAI-WU THEORY

This quadratic interaction theory allows for the strength predictions wherein interaction among stress components can be considered in determining strengths in a biaxial field. Thus, in the case of an orthotropic lamina in a general state of planar stress:

$$F_1 \sigma_1 + F_2 \sigma_2 + F_{11} \sigma_1^2 + F_{22} \sigma_2 + 2F_{12} \sigma_1 \sigma_2 + F_{66} \tau_{12}^2 = \text{FI} \qquad \text{(A-108)}$$

$$F_1 = \frac{1}{X_t} - \frac{1}{X_c} \qquad F_2 = \frac{1}{Y_t} - \frac{1}{Y_c} \qquad F_{11} = \frac{1}{X_t X_c} \qquad F_{22} = \frac{1}{Y_t Y_c} \qquad F_{66} = \frac{1}{S^2} \qquad \text{(A-109)}$$

and F_{12} needs to be determined experimentally, from a biaxial test. However, satisfactory results may be obtained by setting it to zero.

MAXIMUM STRESS

Failure is assumed to occur when any one of the stress components is equal to its corresponding intrinsic strength property. In mathematical form, the Maximum Stress theory is given by:

$$\sigma_1 \geq X_t,\ \sigma_1 > 0\ ;\ \sigma_1 \leq -X_c,\ \sigma_1 < 0$$

$$\sigma_2 \geq Y_t,\ \sigma_2 > 0\ ;\ \sigma_2 \leq -Y_c,\ \sigma_2 < 0 \qquad \text{(A-110)}$$

$$\tau_{12} \geq S,\ \tau_{12} > 0\ ;\ \tau_{12} \leq -S,\ \tau_{12} < 0$$

where the intrinsic strength properties are as defined previously.

MAXIMUM STRAIN

The Maximum Strain theory is analogous to the Maximum Stress theory. Failure is assumed to result when any one of the strain components is equal to its corresponding intrinsic ultimate strain. In mathematical form the Maximum Strain theory is given by:

$$\varepsilon_1 \geq E_t,\ \varepsilon_1 > 0\ ;\ \varepsilon_1 \leq -E_c,\ \varepsilon_1 < 0$$

$$\varepsilon_2 \geq F_t,\ \varepsilon_2 > 0\ ;\ \varepsilon_2 \leq -F_c,\ \varepsilon_2 < 0 \qquad \text{(A-111)}$$

$$\gamma_{12} \geq E_s,\ \gamma_{12} > 0\ ;\ \gamma_{12} \leq E_s,\ \gamma_{12} < 0$$

where the intrinsic ultimate strains are as defined previously.

FAILURE INDEX OF BONDING

The failure index of bonding material is calculated as the maximum interlaminar shear stress divided by the allowable bonding stress.

A.10. CORRECTION OF OUT-OF-PLANE SHEAR STRAIN

The typical formulation for a QUAD4 type finite element follows a standard bilinear isoparametric theory, with directional reduced integration for out-of-plane shear strain. However, this formulation has been found to be inadequate when the geometry of the element is irregular, and a correction defined herein has been implemented in ASTROS to correct this problem.

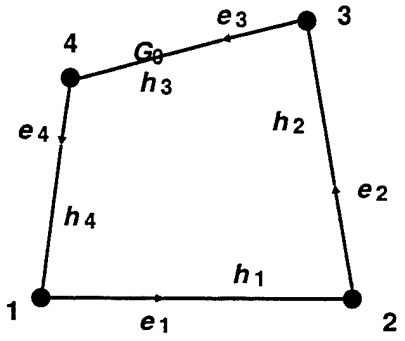
The modification is based upon the theory presented by Hughes and Tezdayar (Reference A-1), but is generalized to include non-planarity of the element, and special features to accommodate the ASTROS structure. The formulation enforces constant shear along each edge of the element, eliminating the need to perform reduced integration.

The formulation of this modification consists of establishing strain-displacement relationships in the element coordinate system. It involves six degrees of freedom (dof), the rotational part of which will be modified later to include the singularity about the normal to the mid-surface.

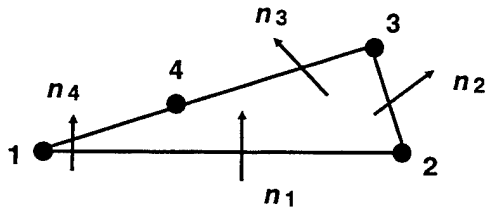
A.10.1. Geometric Variables

The following terms are defined for each edge of an irregular-shaped, non-planar element:

A Unit Normal Vector ($\vec{e}$) in the direction of the next node as illustrated;



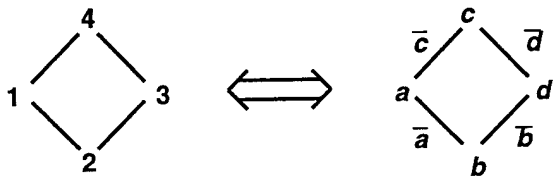
A Unit Normal Vector ($\vec{n}$) which is a normalized average of the nodal normals to the mid-surface along that edge;



Length of each edge $\left(h_i \right)$; and cosine of the internal angle at each corner $\left(\alpha_i \right)$.

A.10.2. Edge Shears and Shear Vectors

Given the following numbering sequence:



At the middle of each edge, the constant shears parallel to edges $\bar{a}$, $\bar{b}$, $\bar{c}$ and $\bar{d}$, respectively, are:

$$
\begin{aligned}
g_{\bar{a}} &= \frac{1}{h_a}\,\vec{n}_a \cdot \left(\vec{U}_b - \vec{U}_a\right) - \frac{1}{2}\,\vec{e}_a\left(\vec{\theta}_b + \vec{\theta}_a\right) \\
g_{\bar{b}} &= \frac{1}{h_b}\,\vec{n}_b \cdot \left(\vec{U}_d - \vec{U}_b\right) - \frac{1}{2}\,\vec{e}_b\left(\vec{\theta}_d + \vec{\theta}_b\right) \\
g_{\bar{c}} &= \frac{1}{h_c}\,\vec{n}_c \cdot \left(\vec{U}_a - \vec{U}_c\right) - \frac{1}{2}\,\vec{e}_d\left(\vec{\theta}_a + \vec{\theta}_c\right) \\
g_{\bar{d}} &= \frac{1}{h_d}\,\vec{n}_d \cdot \left(\vec{U}_c - \vec{U}_d\right) - \frac{1}{2}\,\vec{e}_d\left(\vec{\theta}_c + \vec{\theta}_d\right)
\end{aligned}
\qquad \text{(A-112)}
$$

where $\vec{u}$ and $\vec{\theta}$ are the vectors of translations and rotations at each node, respectively, in the element coordinate system.

The shear vector $\vec{\gamma}_b$ at node (b) is given by:

$$
\vec{\gamma}_b = \frac{1}{1-\alpha_b^2}\left(g_{\bar{b}} + g_{\bar{b}}\,\alpha_b\right)\vec{e}_b + \frac{1}{1-\alpha_a^2}\left(g_{\bar{b}} + g_{\bar{b}}\,\alpha_b\right)\vec{e}_a \qquad \text{(A-113)}
$$

or

$$\vec{\gamma}_b = \left[-\frac{1}{\left(1-\alpha_b^2\right)h_a} \left(\vec{e}_a + \alpha_b \vec{e}_b\right)\left(\vec{n}_a \cdot \vec{U}_a\right) \right] + \left[\frac{1}{\left(1-\alpha_b^2\right)h_a} \left(\vec{e}_a + \alpha_b \vec{e}_b\right)\left(\vec{n}_a \cdot \vec{U}_b\right) \right]$$

$$-\left[\frac{1}{\left(1-\alpha_b^2\right)h_b} \left(\vec{e}_b + \alpha_b \vec{e}_a\right)\left(\vec{n}_b \cdot \vec{U}_b\right) \right] + \left[\frac{1}{\left(1-\alpha_b^2\right)h_b} \left(\vec{e}_b + \alpha_b \vec{e}_a\right)\left(\vec{n}_b \cdot \vec{U}_d\right) \right]$$

$$-\left[\frac{1}{2\left(1-\alpha_b^2\right)} \left(\vec{e}_a + \alpha_b \vec{e}_b\right)\left(\vec{e}_a \cdot \vec{\theta}_a\right) \right] \qquad \text{(A-114)}$$

$$-\left[\frac{1}{2\left(1-\alpha_b^2\right)} \left(\vec{e}_a + \alpha_b \vec{e}_b\right)\left(\vec{e}_a \cdot \vec{\theta}_a\right) + \left(\vec{e}_b + \alpha_b \vec{e}_a\right)\left(\vec{e}_b \cdot \vec{\theta}_b\right) \right]$$

$$-\left[\frac{1}{2\left(1-\alpha_b^2\right)} \left(\vec{e}_b + \alpha_b \vec{e}_a\right)\left(\vec{e}_b, \vec{\theta}_d\right) \right]$$

and similarly for the other nodes, by permutations of the a, b, c and d subscripts.

A.10.3. Nodal Contributions of Shear Strain

The contribution of each node to the total shear strain $\vec{\gamma}_T$ evaluated at an integration point is:

$$\vec{\gamma}_T = \sum_{i=1}^{4} N_i \vec{\gamma}_i \qquad \text{(A-115)}$$

The "pseudo-contribution" of each edge to the total shear strain (G) has the following form:

$$\vec{G}_{\bar{a}} = \frac{N_a}{1-\alpha_a^2}\left(\vec{e}_a + \alpha_a \vec{e}_c\right) + \frac{N_b}{1-\alpha_b^2}\left(\vec{e}_a + \alpha_b \vec{e}_b\right)$$

$$\vec{G}_{\bar{b}} = \frac{N_b}{1-\alpha_b^2}\left(\vec{e}_b + \alpha_b \vec{e}_a\right) + \frac{N_d}{1-\alpha_d^2}\left(\vec{e}_b + \alpha_d \vec{e}_d\right)$$

$$\vec{G}_{\bar{c}} = \frac{N_c}{1-\alpha_c^2}\left(\vec{e}_c + \alpha_c \vec{e}_d\right) + \frac{N_a}{1-\alpha_a^2}\left(\vec{e}_c + \alpha_a \vec{e}_a\right) \qquad \text{(A-116)}$$

$$\vec{G}_{\bar{d}} = \frac{N_d}{1-\alpha_d^2}\left(\vec{e}_d + \alpha_b \vec{e}_b\right) + \frac{N_c}{1-\alpha_c^2}\left(\vec{e}_d + \alpha_c \vec{e}_c\right)$$

Hence, the columns of the $\boldsymbol{B}$ matrix partition for shear, corresponding to node b, $\boldsymbol{BS}_b$, are:

$$\left.\begin{aligned} \overline{\boldsymbol{BS}}_{bi} &= \frac{n_a^i}{h_a}\vec{G}_a - \frac{n_b^i}{h_b}\vec{G}_b \\ \overline{\boldsymbol{BS}}_{bj} &= \frac{e_a^i}{2}\vec{G}_a - \frac{e_b^i}{2}\vec{G}_b \end{aligned}\right\} \quad \begin{aligned} i &= 1, 2, 3 \\ j &= 4, 5, 6 \end{aligned} \tag{A-117}$$

A.10.4. Transformations

The following transformations have to be performed before the preceding formulation can replace the existing $\boldsymbol{B}$ matrix generation for out-of-plane shear.

$$\boldsymbol{BS}_{b\,(3x6)} = \boldsymbol{T}_{IE\,(3x3)}\,\overline{\boldsymbol{BS}}_{b\,(3x6)} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T}_{EE\,(6x6)} \end{bmatrix} \tag{A-118}$$

where

$\boldsymbol{T}_{IE}$ Is the orthogonal transformation between integration points and the element coordinate system, required since all the strains are calculated in the I system.

$\boldsymbol{I}$ Is a 3x3 identity matrix.

$\boldsymbol{T}_{EE}$ Is the 3x3 transformation which takes into account the following facts

A. Hughes' convention for rotations is different than the one implemented in ASTROS; and,

B. The rotation about the normal to the mid-surface at each grid point is singular.

If NV is the normal vector at a given grid point, then:

$$\boldsymbol{T}_{EE} = \begin{bmatrix} 0 & -NV^3 & NV^2 \\ NV^3 & 0 & -NV^1 \\ -NV^1 & NV^1 & 0 \end{bmatrix} \tag{A-119}$$

REFERENCES

A-1 Hughes, T. J. R. and Tezduyar, T. E., "Finite Elements Based Upon Mindlin Plate Theory with Particular Reference to the Four-Node Bilinear Isoparametric Element," Transactions of ASME, Journal of Applied Mechanics, Volume 48, No. 3, September, 1981, pp 587-596.

APPENDIX B. NUCLEAR BLAST ANALYSIS

This appendix provides a theoretical development for the methodology employed in the aerodynamic preprocessor which computes indicial response coefficients for use in the blast response calculations. This appendix was prepared by Robert F. Smiley of Kaman AviDyne.

The basic concept of the Aero preprocessor module of Figure 22 of Section 12 is to obtain the indicial response functions governing the time-history loading response of a given computational box on the aircraft structure to an indicial normal wash at the same or any other box. This problem is addressed by first obtaining the corresponding frequency response functions and then transforming these functions to the time domain by appropriate Fourier transforms (Reference B-1). For the most part the discussion is made with respect to the subsonic case and the doublet-lattice method, but the discussion (with some indicated differences) applies also to the supersonic case and the constant pressure method (CPM).

For low frequencies, the frequency response functions are obtained from the doublet-lattice (or CPM) method as a function of the reduced frequency, k. This procedure is considered valid for practical applications up to some upper frequency limit, k_{max}, which may be estimated by the guidelines of References B-2 for the doublet-lattice case. A typical limit for high subsonic speeds is $k_{max} = 2$.

The initial indicial response, corresponding to an infinite frequency, is obtained using piston theory, which gives a value of lift coefficient slope of $\frac{4}{M}$ (where M is the Mach number) for the effect of a box on itself and zero on other boxes, or in terms of a piston theory pressure P_T:

$$P_T = \frac{4}{M}\,\bar{q}\alpha \tag{B-1}$$

where,

α is the indicial angle of attack

$\bar{q}$ is the dynamic pressure

M is the Mach number

For frequencies between k_{max} and infinity, no practical theoretical methods for calculations were identified. However, it is necessary to estimate (interpolate) the frequency response functions in this intermediate range in order to define the time-domain indicial response coefficients by the Fourier transform with an accuracy sufficient to provide desired time-domain resolution of local blast loading transients. In particular, it may be required to provide time resolution on the order of the diffraction period, corresponding to the time required for a blast wave to cross a lifting surface, or, for higher local

resolution, the time to cross a computational box. To meet this requirement, a semi-empirical interpolation discussed here was adopted.

The first case considered is that of the influence of a sending box on other receiving boxes, where piston theory gives zero at infinite frequency. The doublet-lattice frequencies are extended to infinity from the values calculated for k_{max} assuming the amplitude of the complex frequency-response function varies inversely with frequency as $\frac{A}{k} + \frac{B}{k^2}$ for $k \geq k_{max}$ and that the rate of change of the phase of the complex function with frequency is constant and has the same value as is obtained from the two highest frequencies calculated by the doublet-lattice method. The constants A and B in the amplitude expressions are chosen so that the magnitude and slope of the amplitude function are continuous through k_{max}.

For the case of the response of a box to itself, a two-stage approach is followed to provide a reasonable first approximation to the diffraction loading on the lifting surface. As a first step, the doublet-lattice values are extrapolated to higher frequencies than k_{max} by assuming constant amplitude calculated from the value at k_{max} and constant rate of change of phase angle calculated from the value at k_{max} and the next lower frequency. These results are then inverted into the time domain by the Fourier transform to obtain raw values which are sufficiently accurate for generalized times greater than some minimum time designated s_{dl}, of the order $\frac{1}{k_{max}}$. These raw values are corrected in the time-domain to conform to estimates of indicial response values for times preceding s_{dl}, made as described later (in the discussion of Equation B-4).

For all cases, conversion from frequency domain coefficients to time-domain coefficients is obtained by the Fourier transform expressed in the form:

$$F(s) = \frac{1}{\pi} \int_o^\infty \frac{1}{k} \mathrm{Im}\left[C(k)\, e^{iks} \right] dk \tag{B-2}$$

$$s = \frac{U t}{b} \tag{B-3}$$

where,

C	is the complex frequency response function
$Im\,(\,)$	designates the imaginary part of a function
k	is generalized frequency ($\omega b/U$)
ω	is frequency
s	is generalized time
b	is the reference semi-chord
U	is free stream velocity

Equation B-2 differs from the form used in Reference B-3 in that the complex form of the frequency function is used instead of the real form. This was done because it produced more efficient computations.

Numerical integration of Equation B-2 is performed by assuming that the amplitude and phase angle of the frequency function vary linearly between calculated values of the complex frequency function for the $k \leq k_{max}$ and by using the inverse k expressions previously described from k_{max} to infinity.

The remaining task is to correct the *box-on-itself* time-domain coefficients of Equation B-2 to take into account early time loading (including diffraction effects). Consider a streamwise section of the associated airfoil through a sending box. The total loading contribution L produced by a sending box of area S on all receiver boxes is assumed to be given by the expression:

$$\frac{L}{S} = p_T \quad \text{when} \quad s < s_{arr} \tag{B-4a}$$

$$\frac{L}{S} = p_T \left[\frac{a}{b}\right]^n \quad \text{when} \quad s_{arr} \leq s \leq s_{amm} \tag{B-4b}$$

where the first term represents piston theory and the arrival time s_{arr} is the generalized time required by a sonic signal from the box to reach the leading or trailing edge, whichever is smaller. This is the time up to which Equation B-4a is exactly valid. The upper limit s_{amm} for Equation B-4b is discussed below. The power law form of the second expression was selected because the exponent n can be evaluated from theoretical expressions for the diffraction loading on a two-dimensional airfoil (c.f., Reference B-4). Integration of Equation B-4 over a streamwise strip of an airfoil, assuming an infinite number of chordwise boxes, gives the following expression for the average pressure on the strip due to indicial motion of the entire strip:

$$p_{avg} = p_T \left[1 - \frac{2}{M}\left(\frac{n}{n+1}\right)\frac{b}{c}\, s\right] \quad ; \quad s \leq s_{amm} \tag{B-5}$$

where,

$$s_{amm} = \left(\frac{M}{2}\right)\left(\frac{c}{b}\right)$$

The corresponding expression from linearized two-dimensional subsonic unsteady flow theory (Reference B-4) is:

$$s \leq \frac{M}{1+M}\,\frac{c}{b} \tag{B-6}$$

where

$$C = \frac{1-M}{M}\,\frac{b}{c} \tag{B-7}$$

c is the chord

M is the freestream Mach number

This expression can be modified to take sweep angle (Λ) into account for wing of infinite aspect ratio by calculating the flow process in cross-flow planes perpendicular to the leading edge of the wing, which results in a modification of Equation B-7 to the form:

$$C = \frac{sec\,\Lambda - M}{M}\,\frac{b}{c} \tag{B-8}$$

Since Equations B-5 and B-6 have the same form, they may be compared, using Equation B-8, to solve for the unknown parameter, n, in Equation B-5 giving:

$$n = \frac{sec\,\Lambda - M}{M + 2 - sec\,\Lambda} \tag{B-9}$$

where an average value of $sec\,\Lambda$ is used which is the average of the secant of the sweepback angle for the leading and trailing edges. Using this value of n, Equation B-4 may be now used to calculate total loading produced by any sending box up to the time value s_{amm} when the sonic signals from all sending boxes in this strip have reached either the leading or trailing of the strip.

The above discussion to justify the use of Equation B-4 for calculating total loads at early times is concerned with the two-dimensional case. However, the results may be reasonably applied to the three-dimensional case for total loads produced by a three-dimensional sending box for the following reasons. First, for $s \leq s_{arr}$, Equation B-4a is still exactly valid if s_{arr} is redefined as the minimum time for a sonic signal from the box to reach any lifting surface edge (leading, trailing, tip, root). The implemented preprocessor algorithm does not take into account either tip or sweep effects on s_{arr} since this required more work than appeared justified. Similarly, for the range $s_{arr} < s < s_{amm}$, Equation B-4b is as valid for the three-dimensional case as for the two-dimensional case, aside from effects of sweep and tip/root edges. Note that this applies only to the total load produced by a sending box on a lifting surface. Local loads are, of course, considerably different for two- and three-dimensional cases.

Equation B-4 provides a first approximation for the total loading produced by indicial motion of the box. Since this total loading is distributed over all receiving boxes (including the sending box), the loading on the sending box alone which is consistent with this total loading is obtained by subtracting the calculated loading contribution for all receiver boxes other than the sender box from the total loading.

These results apply mainly to the subsonic case. No detailed study was made of the supersonic case. However, as a first approximation for the supersonic case to be used in conjunction with supersonic constant pressure data (replacing doublet-lattice data in the preceding discussion), the same procedure is used with the following differences. Equation B-4a still applies for less than s_{arr}, where s_{arr} is now

$\frac{M}{(1+M)}\frac{c}{b}$ (Reference B-4). Since there is no linear decay period for the supersonic case corresponding to Equation B-6, the slope C in this equation is taken equal to zero, and s_{amm} is set equal to s_{arr}.

These results define the time-domain response for generalized times less than s_{amm} and for times greater than some value s_{dl} (presently taken as $\frac{2.0}{k_{\max}}$). As stated previously, the raw calculations based on doublet-lattice calculations without piston theory considerations are considered adequate for times after s_{dl}. For intermediate periods, the value of the time-domain coefficients for all sender boxes are extrapolated to meet the raw doublet-lattice curve using the average curve slope given by Equations B-6 and B-8, or (if the curves do not meet) by a straight line connected to the raw doublet-lattice value at s_{dl}.

The procedure presented provides the indicial response function $F(s)$ for all sender receiver box combinations. To apply these functions to practical blast response problems, it is convenient to express this function as a decaying exponential series of the form:

$$F(s) = \alpha_o + \sum_n a_n e^{-\beta_n s} \qquad \text{(B-10)}$$

(as discussed in Reference B-3). The beta parameters β_n are chosen so that the function $F(s)$ in Equation B-10 can be fitted, with a tolerable error, for all generalized times of interest, from zero to infinity, using a minimum number of β coefficients. In recent Kaman AviDyne work seven β values were adequate, having the values:

$$\beta_1 = b_{\min} = 0.375, \ldots, b_{\max} = \beta_7 = 10.0$$

where, values between $b_{\min}$ and $b_{\max}$ are logarithmically spaced. These particular values are provided as default values in the ASTROS program.

Values of the coefficients a_n are obtained by calculating values of $F(s)$ in Equation B-10 for M values of generalized time s_m and then satisfying Equation B-10 by the least-square procedure. The values of s_m are chosen to cover the range of interest. The following values have been used in Kaman AviDyne calculations and are provided as default values in the ASTROS program:

$$M = 20$$

$$s_1 = s_{\min} = 0.1, \ldots, s_{20} = s_{\max} = 21.0$$

where values between $s_{\min}$ and $s_{\max}$ are logarithmically-equally spaced.

REFERENCES

B-1 Garrick, I. E., "On Some Fourier Transforms in the Theory of Non-Stationary Flows," 5th International Congress for Applied Mechanics, 1938, John Wiley & Sons, Inc., 1939, pp. 590-593.

B-2 Giesing, J. P., Kalman, T. P., and Rodden, W. P., "Subsonic Unsteady Aerodynamics for General Configurations," AFFDL-TR-71-5, Part I, Volume I - Direct Application of the Nonplanar Doublet-Lattice Method; Part I, Volume II - Computer Program H7Wc; Part II, Volume * - Application of the Doublet-Lattice Method and the Method of Images to Lifting-Surface/Body Interference; Part II, Volume II - Computer Program N5KA; Volume I, November 1971, Volume II, April 1972.

B-3 Zartarian, G., "Application of the Doublet-Lattice Method for Determination of Blast Loads on Lifting Surfaces at Subsonic Speeds," AFWL-TR-72-207, 1973.

B-4 Lomax, H., et al, "Two- and Three-Dimensional Unsteady Lift Problems in High-Speed Flight," NACA Report 1077, 1952.

APPENDIX C. THE FAST FOURIER TRANSFORM

The equation of motion for a transient response dynamic analysis is given by:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{B}\dot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{P}(t) \tag{C-1}$$

Where $\boldsymbol{M}$, $\boldsymbol{B}$ and $\boldsymbol{K}$ are the mass matrix, the damping matrix and the stiffness matrix, respectively. $\boldsymbol{P}(t)$ is the external load vector in the time domain, and $\boldsymbol{u}$ is the response displacement vector. Here, Equation C-1 is assumed reduced to the solution set degrees-of-freedom. In general, Equation C-1 can be solved by numerical integration. However, another method for solving this equation is the Fourier Transform (FT) technique. In this method, Equation C-1 is first transformed into the frequency domain with the Fourier Transform, the response displacement vector is computed in the frequency domain, and finally the frequency domain displacement vector is transformed back into the time domain by using the Inverse Fourier Transform.

In general, an external load vector $\boldsymbol{P}(t)$ can be transformed to the frequency domain using:

$$\boldsymbol{P}(\omega) = \int_o^\infty \boldsymbol{P}(t)\, e^{-i\omega t}\, dt \tag{C-2}$$

Equation C-1 in the frequency domain is then

$$\left[-\omega^2 \boldsymbol{M} + i\,\omega \boldsymbol{B} + \boldsymbol{K}\right] \bar{\boldsymbol{u}} = \boldsymbol{P}(\omega) \tag{C-3}$$

After the displacement vector $\bar{\boldsymbol{u}}$ is obtained by the frequency response method, it can be transformed back into the time domain by

$$\boldsymbol{u}(t) = \frac{1}{\pi} \int_o^\infty \left(\bar{\boldsymbol{u}}(\omega)\, e^{i\omega t}\right) d\omega \tag{C-4}$$

For certain types of problems, the use of the Fourier Transform method offers many advantages over numerical integration methods. For periodic external load vectors, the FT method can be used to obtain the accumulated effects on the response displacement vector and, at the same time, minimize the computing costs. Even for nonperiodic external loads, this method may be more efficient than a numerical integration approach. It should be stressed that while the Fourier Transform is presented here in terms of performing transient response of structures, the method has wide applicability. Therefore, the availability of FT-based algorithms in ASTROS provides a building block for other disciplines.

C.1. DISCRETE FOURIER TRANSFORM

There are two difficulties in practical applications of the Fourier Transform method as described by Equations C-2 through C-4. First, the time function P(t) is continuous; however, in practice, its values are known only at a finite number of time points. Second, while the integration limits in Equation C-2 are from zero to infinity, practical applications must have a finite time duration. Therefore, the Fourier Transform method needs to be reformulated such that it can be managed practically. This form is called the Discrete Fourier Transform (DFT). A summary of the theory of the DFT is given here while References C-1 and C-2 provide more detailed information.

For a function $P(t)$ defined over the time duration T and with N sample points at which the values of the function $P(t)$ are known, i.e.,

$$P_n = P(t_n) \tag{C-5}$$

where

$$t_n = n\Delta t, \quad n = 0, 1, 2, ..., N$$

$$\Delta t = \frac{T}{N}$$

three important parameters in the frequency domain can be derived: the incremental frequency, Δf_f, the number of frequency steps N_f, and the frequency duration, F_f:

$$\Delta f_f = \frac{1}{T}$$

$$N_f = \frac{N}{2} \tag{C-6}$$

$$F_f = N_f \Delta f_f$$

A key requirement for the discrete transform to be valid is that the excitation be periodic "in the window," i.e., the time duration T. This is a rather special case, but a wider range of cases can be considered by recognizing that a response that dies out within the window could be considered periodic since the responses in successive periods are uncoupled. Clearly, for the response to die out, the excitation must be zero for the last portion of the period T.

The *DFT* and *IDFT* (Inverse *DFT*) can now be defined as:

$$F(\omega_m) = \frac{1}{N} \sum_{n=0}^{N-1} f(t_n) e^{-i\omega_m t_n} \tag{C-7}$$

$$f(t_n) = F(0) + 2 \sum_{m=1}^{N_f - 1} R_e \left[F(\omega_m) e^{i \omega_m t_n} \right] \tag{C-8}$$

where

$$\omega_m = 2\pi \Delta f_f m$$
$$t_n = n \Delta t \tag{C-9}$$

$F(\omega_m)$ is typically complex although the $F(0)$ term is seen, from Equation C-7, to be real.

C.2. FAST FOURIER TRANSFORMS

The evaluation of the *DFT* and the *IDFT* of a function $P(t)$ as given by Equations C-7 and C-8 are accomplished by a numerical technique which is known as the Fast Fourier Transform (FFT). This procedure is very powerful in that it reduces the number of multipliers to compute the transformed quantity from $\frac{N^2}{2}$ to $N \log_2 N$. Figure C-1 shows a comparison of computation times for FFT and a brute force approach.

The following is a conceptual description of the FFT. Additional details are found in References C-2 and C-3. The restriction is first made that the number of time points is a power of two.

$$N = 2^M \tag{C-10}$$

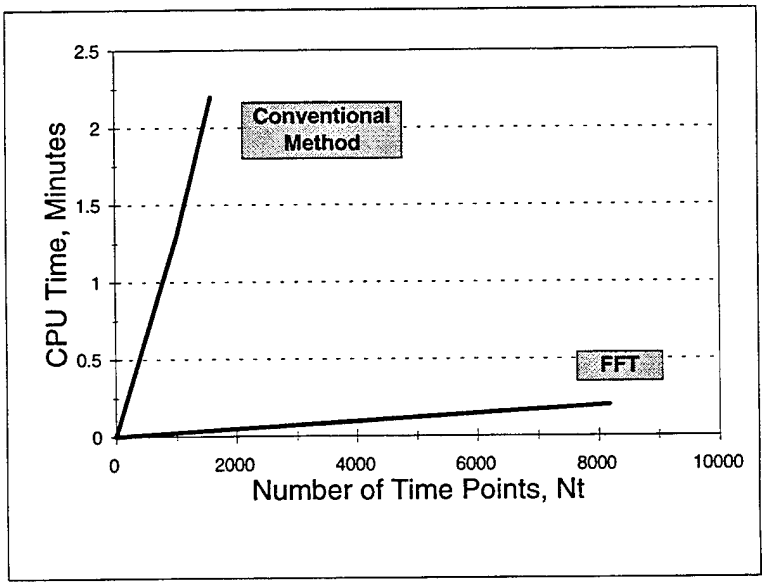


Figure C-1. CPU Time Comparison

where M is an integer. The transform of Equation C-7 can be written as:

$$F(m) = \sum_{n=0}^{N=1} f(n)\ W_n^{mn} \tag{C-11}$$

where $W_N = e^{\frac{-2\pi i}{N}}$ and the relations of Equation C-6 have been used to replace $\omega_m t_n$ with $\frac{(2\pi mn)}{N}$. The integers m and n can be expressed in binary form as:

$$\begin{aligned} m &= m_o + 2m_1 + 4m_2 \ldots + 2^{m-2}\ m_{m-2} \\ n &= n_o + 2n_1 + 4n_2 \ldots + 2^{M-1}\ n_{M-1} \end{aligned} \tag{C-12}$$

where the values of m_i are either 0 or 8. For illustrative purpose, set $N = 8$, so that $M = 3$, then

$$\begin{aligned} m &= m_o + 2m_1 \\ n &= n_o + 2n_1 + 4n_2 \end{aligned} \tag{C-13}$$

and Equation C-11 becomes

$$F(m) = \sum_{n_2=0}^{1} \sum_{n_1=0}^{1} \sum_{n_0=0}^{1} f(n)\ W_8^{(m_o+2m_1)(n_o+2n_1+4n_2)} \tag{C-14}$$

If the exponential term is factored by powers of two then

$$W_8^{mn} = W_8^{8m_1n_2}\ \ W_8^{4m_0n_2}\ \ W_8^{2m_1\left(m_0+2m_1\right)}\ \ W_8^{n_0\left(m_0+2m_1\right)} \tag{C-15}$$

The first factor on the right hand side is unity since

$$W_8^{8I} = e^{-\frac{2\pi i}{8}8I} = 1 \tag{C-16}$$

where I is an integer. For the remaining terms, the n_i coefficients are segregated so that three intermediate summations can be defined as:

$$F^1(m_0, n_1, n_0) = \sum_{n_2=0}^{1} f(n_2, n_1, n_0)\ W_8^{4n_2m_0} \tag{C-17}$$

where $f(n_2, n_1, n_0)$ is $f(n)$ in Equation C-14. Note that each of the eight terms on the left-hand side is computed from two multiply operations. In a similar fashion, a second summation is

$$F^2(m_0, m_1, n_0) = \sum_{n_1=0}^{1} F^1(m_0, m_1, n_0)\, W_8^{2n_1(m_0+2m_1)} \tag{C-18}$$

and the third and final summation is

$$F^3(m_0, m_1) = \sum_{n_0=0}^{1} F^2(m_0, m_1, n_0)\, W_8^{n_0(2m_1+m_0)} \tag{C-19}$$

This final term is the $F(m)$ of Equation C-14. While the process has been shown for $N = 8$, it does generalize to $N = 2^M$.

C.3. IMPLEMENTATION CONSIDERATIONS

To control the solution of a transient response problem using the FFT, two sets of parameters must be input by the user. The first set contains the parameters used to control the FFT, which are: T, the total time duration and N, the number of time points. With T and N determined, the characteristics in the frequency domain are given by Equation C-6. The time points and their corresponding frequency list are given by Equation C-9.

The second set of parameters is the frequency list used in solving Equation C-3 to obtain the response vector $\boldsymbol{u}(\omega)$. This frequency list is:

$$\omega = 2\pi f_0, 2\pi f_1, 2\pi f_2, \ldots, 2\pi f_n \tag{C-20}$$

The frequency lists ω_m and ω are not necessarily equal. While setting $\omega_m = \omega$ will give the most accurate response, it may not be efficient. To give the user complete control over accuracy versus efficiency, two alternative methods are used to input the frequency list ω. For the first method, the frequency list ω is input via two parameters:

$$R_{\Delta f} = \frac{\Delta f}{\Delta f_f}$$

$$R_f = \frac{F}{F_f} \tag{C-21}$$

where Δf and Δf_f are incremental frequencies used in the frequency response analysis and the FFT, respectively. And F and F_f are total frequency durations used in the frequency response analysis and the FFT, respectively. If $R_{\Delta f} = 1.0$ and $R_f = 1.0$, then the frequency lists ω and ω_m are equal, which gives the most accurate response. For greater efficiency, $R_{\Delta f}$ values greater than one may be used.

The frequency list ω determined by Equation C-21 has an equal frequency interval. In some cases, the user may desire a frequency list with an unequal interval. Therefore, an input option is provided to allow the user to input a completely independent frequency list ω.

If the frequency lists ω and ω_m are not equal, the differences between these two lists are reconciled by either linear or cubic interpolation, with linear interpolation the default.

REFERENCES

C-1. Bracewell, R.N., *The Fourier Transform and Its Applications*, McGraw-Hill Book Company, 1982.

C-2. Elliott, D.F. and Rao, K.R., *Fast Fourier Transform*, Academic Press, Inc., 1982.

C-3. Cooley, J.W., Lewis, P.W., and Welch, P.D., "The Fast Fourier Transform and Its Applications," IEEE Trans. Education, Vol. E-12, No. 1, pp 27-34, March 1969.